
Dedicace

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

À LA MÉMOIRE DE MON PÈRE

À la mémoire de mon père, qui m'a toujours inspiré et encouragé. Son amour, son soutien et ses précieux conseils m'ont guidé tout au long de ma vie. Que ce travail soit une humble reconnaissance de tout ce qu'il a fait pour moi. Puisse Dieu, le tout-puissant, l'accueillir en sa sainte miséricorde et lui accorder le repos éternel.

À MA CHÈRE MÈRE

À ma chère mère, pour votre amour sans limite et vos nombreux sacrifices. Votre soutien indéfectible m'a guidé tout au long de mon parcours. Merci pour tout ce que vous avez fait pour moi. Que ce travail soit une preuve de ma gratitude et de mon amour éternel. Que Dieu vous accorde santé, bonheur et longue vie.

À MES FRÈRES MOTEZ ET BADER

Pour votre soutien et vos encouragements malgré la distance qui nous sépare. Votre présence, même éloignée, a toujours été une source de force et de motivation.

À MA SŒUR SARRA

Pour ta présence, ton amour et ton soutien constant. Que ce travail soit une marque de gratitude pour tout ce que tu représentes pour moi.

À TOUS MES CHERS AMIS

Pour leur aide et leur soutien dans les moments difficiles. Votre amitié est inestimable et a été une source de réconfort et de motivation.

Kamel Jemmali

Dedicace

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

À MON PÈRE

Ce travail est dédié à mon père, qui m'a toujours poussé et motivé dans mes études. J'espère que ce geste modeste soit une preuve de reconnaissance de la part d'un fils qui a toujours prié pour sa réussite et son bonheur. Puisse Dieu, le tout-puissant, l'avoir en sa sainte miséricorde !

À MA CHÈRE MÈRE

Aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération pour les sacrifices que vous avez consentis pour mon instruction et mon bien-être. Je vous remercie pour tout le soutien et l'amour que vous me portez depuis mon enfance et j'espère que votre bénédiction m'accompagne toujours. Que ce modeste travail soit l'exaucement de vos vœux tant formulés, le fruit de vos innombrables sacrifices. Puisse Dieu, le Très-Haut, vous accorder santé, bonheur et longue vie.

À TOUTE LA GRANDE FAMILLE

Avec mes sincères vœux de bonheur, de réussite et de prospérité, je vous dédie ce travail comme preuve d'amour.

À TOUS MES CHERS AMIS

Pour leur aide et leur soutien dans les moments difficiles.

À TOUS MES ENSEIGNANTS

Leur générosité et leur soutien m'obligent à leur témoigner mon profond respect et ma loyale considération.

Yassine Chouchène

Remerciements

Nous tenons tout d'abord à exprimer nos sincères remerciements à l'entreprise ERNST pour avoir accueilli notre groupe et offert l'opportunité de réaliser notre projet au sein de leurs installations. Leur soutien et leur expertise ont grandement contribué à la réussite de notre travail. Nous voudrions également exprimer ma gratitude envers notre superviseur universitaire, M. Walid Hassairi, pour son encadrement précieux et ses conseils avisés tout au long de ce projet. En outre, nous souhaitons remercier notre superviseur d'entreprise, M. Marwen Khouaja, pour sa disponibilité et sa générosité en partageant son expérience professionnelle avec nous. Enfin, nos remerciements vont également à tous ceux qui ont contribué de près ou de loin à la réalisation de ce projet. Leur soutien moral et technique a été précieux et nous a permis d'atteindre nos objectifs avec succès. Merci encore à tous ceux qui ont rendu cette expérience possible. Votre soutien inestimable restera gravé dans nos souvenirs.

Table des matières

Introduction générale	1
1 Présentation générale du projet	3
1.1 Introduction	3
1.2 Contexte et cadre du projet	3
1.2.1 Contexte du projet	3
1.2.2 Problématique	4
1.2.3 Cadre du projet	5
1.3 Présentation de l'organisme d'accueil	6
1.4 Etude de l'existant :	6
1.4.1 Services Existants	6
1.4.2 Description De l'Existant	7
1.4.3 Critiques De l'Existant	7
1.4.4 Solution Proposée	8
1.4.5 Comparaison avec la Solution Existante	8
1.5 Méthodologie adoptée	9
1.5.1 Comparaison de certaines méthodologies	9
1.5.2 Choix de la méthodologie Scrum	11
1.5.3 Processus Scrum	11
1.5.3.1 Processus Scrum Simplifié	12
1.6 Langage de modélisation UML	13
1.6.1 Définition d'UML (Unified Modeling language)	13

1.6.2	Les principaux objectifs de l'utilisation de l'UML	13
1.6.3	Les Différents Diagrammes UML	14
1.7	Conclusion	14
2	Planifications et spécifications des besoins	15
2.1	Introduction	15
2.2	Analyse et spécification des besoins	15
2.2.1	Les besoins fonctionnelles	15
2.2.2	Les besoins non fonctionnelles	16
2.3	Identification et structuration de cas d'utilisation	17
2.3.1	Identification des acteurs du système	17
2.3.2	Diagramme des Cas d'utilisations global	18
2.4	Pilotage du projet avec scrum	20
2.4.1	Equipe Et Rôle	20
2.4.2	Backlog du produit	20
2.4.3	Planification des sprints	22
2.5	Environnement de travail	22
2.5.1	Environnement Matériel	22
2.5.2	Environnement Logiciel	23
2.6	Architecture de l'application	26
2.6.1	Architecture logique	26
2.6.2	Architecture Logicielle	27
2.7	Conclusion	29
3	Etude et réalisation du sprint1	30
3.1	Introduction	30
3.2	Sprint 1 : « Application Mobile »	30
3.2.1	Backlog du Sprint 1	30

3.2.2	Spécification fonctionnelle du Sprint 1	31
3.2.2.1	Diagramme de cas d'utilisation du Sprint 1	31
3.2.2.2	Description textuelle de cas d'utilisations	32
3.2.2.3	Diagramme de classe du Sprint 1	32
3.2.2.4	Diagramme de séquence	34
3.2.3	SignalR	38
3.2.3.1	Fonctionnement de SignalR	39
3.2.4	Reconnaissance Faciale	40
3.2.4.1	Les Relations entre l'Intelligence Artificielle, la machine learning, le deep learning et la data science	41
3.2.4.2	Algorithme utilisé	44
3.2.5	Réalisation du Sprint 1	45
3.3	Conclusion	48
4	Etude et réalisation du sprint2	49
4.1	Introduction	49
4.2	Sprint 2 : Agent Saisie	49
4.2.1	Backlog du Sprint 2	49
4.2.2	Spécification fonctionnelle du Sprint2	50
4.2.2.1	Diagramme de cas d'utilisation du Sprint 2	50
4.2.2.2	Description textuelle de cas d'utilisation	51
4.2.2.3	Diagramme de classe du Sprint 2	53
4.2.2.4	Diagramme de séquence	54
4.2.3	Réalisation du Sprint 2	56
4.3	Conclusion	59
5	Etude et réalisation du sprint3	60
5.1	Introduction	60

5.2	Sprint 3 : « Agent Stock »	60
5.2.1	Backlog du Sprint 3	60
5.2.2	Spécification fonctionnelle du Sprint 3	61
5.2.2.1	Diagramme de cas d'utilisation du Sprint 3	62
5.2.2.2	Description textuelle de cas d'utilisation	62
5.2.2.3	Diagramme de classe du Sprint 3	63
5.2.2.4	Diagramme de séquence	64
5.2.3	Réalisation du Sprint 3	67
5.3	Conclusion	71
6	Etude et réalisation du sprint4	72
6.1	Introduction	72
6.2	Sprint 4 : Administrateur	72
6.2.1	Backlog du Sprint 4	72
6.2.2	Spécification fonctionnelle du Sprint	73
6.2.2.1	Diagramme de cas d'utilisation du Sprint 4	74
6.2.2.2	Description textuelle de cas d'utilisation	74
6.2.2.3	Diagramme de classe du Sprint 1	75
6.2.2.4	Diagramme de séquence	76
6.2.3	Réalisation du Sprint 4	79
6.3	Conclusion	80
	Conclusion générale	81
	Perspectives	83

Table des figures

1.1	Liste des ventes d'application ancienne.	5
1.2	pointage d'application ancienne	5
1.3	Le logo de ERNST	6
1.4	Le diagramme Scrum simplifié	12
1.5	Le logo UML	13
2.1	diagramme de cas d'utilisation globale	19
2.2	MVC WEB	28
3.1	Diagramme de cas d'utilisation pour sprint 1	31
3.2	Diagramme de classe pour le sprint 1	33
3.3	Diagramme de séquence « login du sprint 1 »	35
3.4	Diagramme de sequence « Ajouter une Vente »	36
3.5	Diagramme de sequence « Pointer »	38
3.6	Appels De Procédures à Distance En SignalR	39
3.7	Reconnaissance faciale	40
3.8	ai structure	41
3.9	Interface login	45
3.10	Interface Pointage d'application mobile	46
3.11	Interface AjouterVente pour l'application Mobile	47
3.12	Interface de la liste des ventes pour l'application Mobile	47
3.13	Interface du compte d'utilisateur	48
4.1	Diagramme de cas d'utilisation pour « Agent Saisie »	50
4.2	télécharger la liste des ventes et pointage (excel-pdf)	53

4.3	diagramme de classe du sprint 2	54
4.4	Diagramme de sequence valider/refuser vente	55
4.5	Interfaces pour la liste des ventes	56
4.6	Interfaces pour la liste des pointages du jour et pointage details	57
4.7	Interfaces controle vente	58
5.1	Diagramme de cas d'utilisation d'un agent stock	62
5.2	Diagramme de classe « Agent Stock »	64
5.3	Diagramme de séquence affecter stock	66
5.4	Diagramme de séquence entrer stock	67
5.5	Interface d'entré stock	68
5.6	Interface de l'état stock saisie	68
5.7	Interface de l'état stock terrain	69
5.8	Interface de l'affectation du stock	70
5.9	Interface de liste des articles	70
6.1	Diagramme de cas d'utilisation du sprint 4	74
6.2	Diagramme de classe du sprint 4	75
6.3	Diagramme de séquence pour l'authentification	76
6.4	Diagramme de sequence pour la suppression d'une vente	77
6.5	Diagramme de sequence pour ajouter utilisateur	78
6.6	Interface pour la liste des utilisateurs	79
6.7	Interface de dceclaration des pertes	79

Liste des tableaux

1.1	Tableau comparatif des méthodologies de gestion de projet	10
2.1	Les acteurs et leurs descriptions	18
2.2	Rôles Scrum et personnes affectées	20
2.3	Le backlog Produit	22
2.4	Environnement Matériel	23
3.1	Tableau des histoires et estimations en jours du sprint 1	31
3.2	Description textuelle de cas d'utilisation «Pointer»	32
3.3	Description textuelle de cas d'utilisation « Ajouter une Vente »	32
4.1	Tableau des histoires et estimations en jours du deuxième sprint	50
4.2	Refuser une vente	51
4.3	Valider une vente	51
4.4	Controler le pointage	51
5.1	Tableau des histoires et estimations en jours du sprint 3	61
5.2	Ajouter un article	62
5.3	Affecter un stock	63
5.4	Gérer les Transactions	63
6.1	Tableau des histoires et estimations en jours du sprint 4	73
6.2	Ajouter un compte	74

Introduction générale

Dans le monde moderne, la transformation digitale a atteint des sommets inégalés, remodelant la manière dont les organisations opèrent et interagissent avec leur environnement. Dans cette ère numérique en constante évolution, la modernisation des outils technologiques devient impérative pour rester compétitif et efficace. C'est dans ce contexte que s'inscrit notre projet de fin d'études intitulé "Advasim", qui vise à moderniser à la fois le site web existant et l'application mobile utilisée par les agents sur le terrain.

Actuellement, le site web constitue une pièce maîtresse dans le paysage opérationnel des agents centraux, offrant des fonctionnalités cruciales telles que la consultation des ventes, la validation et le refus des ventes, la gestion des stocks et la vérification des puces. Il sert de pivot central pour les opérations quotidiennes, contribuant de manière significative à la gestion fluide et efficace des activités commerciales.

Cependant, avec l'évolution rapide de la technologie, il est devenu impératif de repenser et de revitaliser cette plateforme pour rester à la pointe de l'efficacité opérationnelle. De même, l'application mobile utilisée par les agents sur le terrain est une composante essentielle de notre écosystème opérationnel. Elle permet actuellement la réalisation de ventes sur place, offrant une flexibilité et une agilité inestimables aux agents travaillant directement avec les clients. En outre, les agents de terrain peuvent réclamer une perte et effectuer le pointage, ajoutant une couche supplémentaire de contrôle et de responsabilité. Dans notre quête d'innovation continue, nous cherchons à repousser les limites de ces fonctionnalités pour répondre aux exigences changeantes du marché et des clients.

L'objectif central de cette refonte ambitieuse est de simplifier et d'améliorer de manière significative les processus de vente, de pointage et de gestion des stocks à travers ces applica-

tions modernisées. Nous nous engageons à créer une expérience utilisateur optimale avec une interface intuitive, des fonctionnalités robustes et une intégration transparente entre le site web central et l'application mobile sur le terrain. Cette modernisation vise à accroître l'efficacité opérationnelle, à réduire les délais et à fournir aux agents les outils nécessaires pour exceller dans leurs missions.

Ce rapport est structuré en six chapitres. Le premier chapitre, "Présentation générale du projet", introduit le projet Advasim, ses objectifs et son contexte. Le deuxième chapitre, "Analyse, Méthodologie et Environnement", présente l'approche méthodologique adoptée, notamment le framework Scrum, ainsi que l'environnement matériel et logiciel utilisé. Le troisième chapitre, "Agent terrain et superviseur", se concentre sur les fonctionnalités et les améliorations apportées aux outils utilisés par les agents sur le terrain et leurs superviseurs, y compris la capacité de réclamer une perte et de pointer. Le quatrième chapitre, "Agent de saisie" compris la capacité de valider ou refuser les ventes, ainsi que de consulter la liste des pointages. Le cinquième chapitre, "Agent de stock", traite de la gestion des stocks et des fonctionnalités associées. Enfin, le sixième chapitre, "Administrateur", aborde les outils et fonctionnalités spécifiques à l'administration du système.

En conclusion, notre projet Advasim se positionne comme une réponse innovante aux défis actuels de la digitalisation des processus commerciaux, visant à offrir une solution technologique moderne, efficace et intégrée.

Chapitre 1 : Présentation générale du projet

1.1 Introduction

Ce chapitre est dévolu à la présentation générale de l'entreprise d'accueil et le cadre général du projet. Ensuite nous présentons l'étude de l'application existante et nous introduisons la solution proposée.

1.2 Contexte et cadre du projet

1.2.1 Contexte du projet

L'objectif de notre projet, intitulé "Advasim", est de moderniser la gestion des ventes de puces téléphoniques pour améliorer le contrôle et l'efficacité des opérations commerciales. Nous développerons une application web et une application mobile hybride pour optimiser les processus de vente, de gestion des stocks et de contrôle. Ces applications offriront une gestion en temps réel, la reconnaissance faciale pour le pointage, une interface utilisateur intuitive et une intégration transparente entre les plateformes. L'accent sera mis sur l'automatisation des tâches répétitives pour accroître l'efficacité. Notre projet vise à simplifier les opérations, augmenter la productivité des agents et améliorer l'expérience client en fournissant des services rapides et fiables.

1.2.2 Problématique

Notre application actuelle, pivotale pour nos opérations quotidiennes, est confrontée à plusieurs défis critiques qui nuisent à notre efficacité et restreignent notre capacité à satisfaire les attentes du marché. Premièrement, l'application, développée initialement avec Ionic et ASP.NET MVC Core 3.1, subit des lenteurs notables qui impactent directement la réactivité de nos agents.

Deuxièmement, la gestion des stocks, en raison d'une intégration défaillante, compromet la fiabilité du suivi en temps réel des inventaires.

Troisièmement, notre système souffre de limitations en termes de compatibilité multiplateforme avec les systèmes Android et iOS, ce qui limite notre portée auprès des utilisateurs.

En outre, l'absence de transmission de données de géolocalisation pour chaque vente ou pointage réduit le contrôle et le suivi des agents de terrain.

Enfin, le système de pointage de la première version de l'application permettait aux agents de soumettre n'importe quelle image lors du pointage, posant ainsi des problèmes de vérification de l'identité. Ces problématiques doivent être adressées pour améliorer notre compétitivité et notre efficacité opérationnelle.

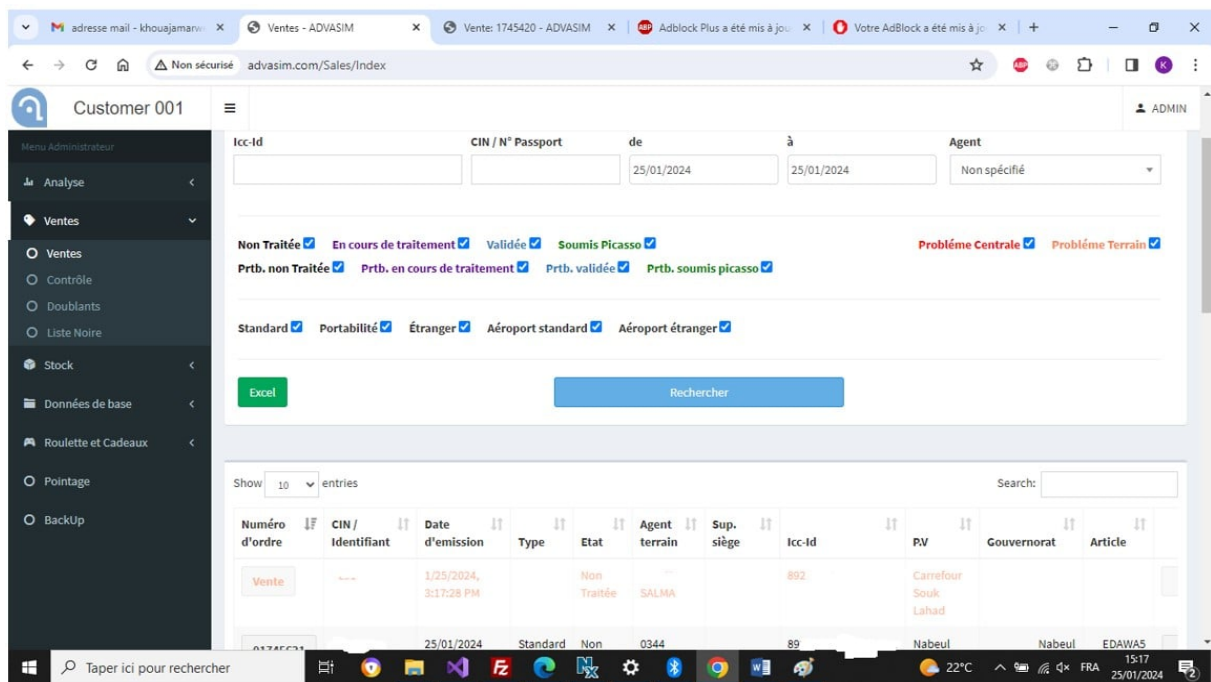


Figure 1.1 – Liste des ventes d’application ancienne.

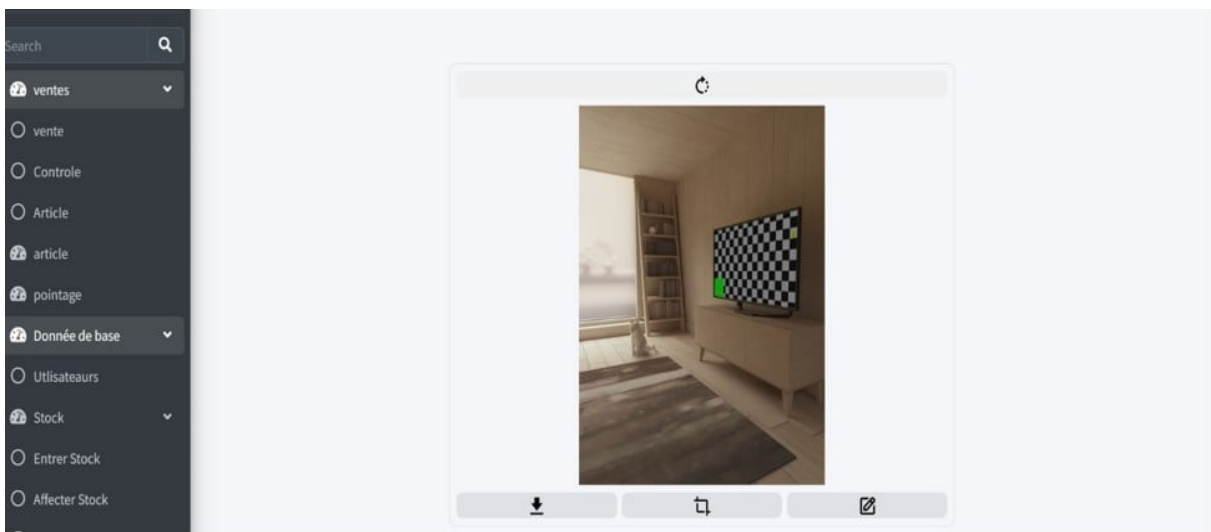


Figure 1.2 – pointage d’application ancienne

1.2.3 Cadre du projet

Ce stage se situe dans le cadre d’un projet de fin d’études a fin de l’obtention du diplôme de la Licence appliquée en informatique, spécialité : Génie Logiciel et système d’information. Il est effectué au sein de la société ERNST qui est spécialisée dans le domaine de développement

des solutions web et mobile. Ce projet consiste à concevoir et développer une plateforme (site + une application mobile hybride) de gestion des puces.

1.3 Présentation de l'organisme d'accueil

ERNST a été fondée en 2013. Elle se présente comme étant une société spécialisée dans le développement des logiciels web, mobile, bureaux et systèmes informatique. Cette dernière offre à ses clients un très haut niveau de qualité de service avec sa composition juvénile. Elle bénéficie d'un grand dynamisme, d'un esprit compétitif et surtout beaucoup d'innovation.



Figure 1.3 – Le logo de ERNST

La société ERNST offre une variété de services tel que :

- Le développement web et mobile.
- La maintenance des systèmes.
- L'administration des serveurs.

1.4 Etude de l'existant :

1.4.1 Services Existants

Site Web Central : Utilisé par les agents centraux pour consulter les ventes, valider les transactions, gérer les stocks et vérifier les puces.

Application Mobile : Utilisée par les agents de terrain pour réaliser des ventes sur place, soumettre des pointages et réclamer des pertes.

1.4.2 Description De l'Existant

Le site web central et l'application mobile offrent des fonctionnalités de base, mais présentent des limitations notables :

Site Web Central : Bien qu'il permette la consultation des ventes, la validation et le refus des ventes, la gestion des stocks et la vérification des puces, il manque d'ergonomie et de modernité. Les agents doivent naviguer à travers plusieurs étapes pour accomplir des tâches simples, ce qui peut être fastidieux et source d'erreurs.

Application Mobile : Permet la réalisation de ventes sur place, mais n'intègre pas la géolocalisation ni la reconnaissance faciale pour le pointage. Cette application, bien qu'utile, est limitée en termes de fonctionnalités et d'intégration avec le site web central, créant des silos d'information et compliquant la synchronisation des données.

1.4.3 Critiques De l'Existant

Les critiques principales des systèmes actuels incluent :

Interfaces obsolètes et non intuitives : Rendant l'utilisation quotidienne difficile et inefficace.

Fonctionnalités limitées et intégration insuffisante : Les systèmes ne communiquent pas efficacement entre eux, ce qui entraîne des inefficacités opérationnelles.

Inefficacité et retard dans les opérations : Les processus ne sont pas optimisés, ce qui entraîne des retards et des erreurs dans les opérations.

Manque de flexibilité : Les systèmes actuels ne s'adaptent pas bien aux évolutions des besoins des utilisateurs et aux exigences du marché.

1.4.4 Solution Proposée

Pour résoudre ces défis, nous proposons une modernisation complète de notre site web et de notre application mobile. Les principales composantes de notre solution sont :

Nouvelle architecture robuste : Améliorer la réactivité et la performance globale en utilisant .NET Core MVC 8 et Flutter pour une meilleure intégration et une expérience utilisateur améliorée.

Surveillance proactive et vérification : Intégrer des systèmes avancés pour une surveillance continue et des contrôles de vérification plus efficaces, minimisant les risques de fraude.

Géolocalisation intégrée : Intégrer la géolocalisation pour chaque vente et pointage, permettant un suivi précis des agents sur le terrain. Gestion agile des stocks : Adopter des méthodes de gestion des stocks plus flexibles et réactives pour s'adapter rapidement aux changements de demande et éviter les surplus ou les pénuries.

Reconnaissance faciale : Améliorer la vérification de l'identité en intégrant la technologie de reconnaissance faciale, restreignant les soumissions aux images faciales des agents et améliorant la sécurité des procédures de pointage.

1.4.5 Comparaison avec la Solution Existante

La plateforme initiale, basée sur .NET Core MVC 3.1 et Ionic, utilisait des technologies web (HTML, CSS, JavaScript) dans un "webview", ce qui peut réduire les performances par rapport aux applications natives. La mise à jour vers .NET Core MVC 8 et Flutter offre des améliorations significatives :

- **.NET Core MVC 8** : Fournit des améliorations en termes de performance et de sécurité, avec une architecture plus robuste pour le site web.
- **Flutter** : Permet le développement d'applications natives pour Android et iOS, garantissant une meilleure intégration système et une expérience utilisateur optimisée.

Cette mise à niveau assure des applications plus performantes et esthétiquement attrayantes,

renforçant notre compétitivité et notre capacité à répondre aux attentes des utilisateurs dans un environnement technologique en constante évolution.

1.5 Méthodologie adoptée

Lors de la mise en œuvre d'un projet informatique, choisir une méthodologie appropriée est crucial pour guider le processus de développement et assurer la réussite du projet et minimiser la perte de temps.

1.5.1 Comparaison de certaines méthodologies

	Description	Points Forts	Points Faibles
Waterfall (En cascade)	Le modèle en cascade, considéré comme l'un des piliers du développement de projets, demeure une approche largement adoptée dans de nombreux secteurs. Son application s'étend à une diversité de projets, allant du développement logiciel à la gestion de projets d'ingénierie.	- Facile à comprendre et à gérer, surtout pour les projets simples et de petite taille où les exigences sont claires dès le début. - Bien adaptée pour les environnements réglementés où des étapes précises et une documentation extensive sont nécessaires.	- Manque de flexibilité pour adapter ou modifier le projet une fois qu'une phase est terminée. - Risque élevé que le produit final ne réponde pas aux attentes des utilisateurs si leurs besoins évoluent au cours du développement.

Extreme Programming (XP)	La méthodologie Extreme Programming (XP) est une approche de gestion de projet qui intensifie les principes du développement agile. Cette méthode met l'accent sur la satisfaction des besoins clients, en adoptant un développement itératif et une intégration continue. Au cœur de XP se trouve la dynamique de l'équipe de projet et ses interactions avec le client. Les phases principales de la méthodologie XP incluent la planification, la conception, la programmation et les tests.	<ul style="list-style-type: none"> - Améliore la qualité du logiciel et la satisfaction du client grâce à l'intégration continue et au feedback rapide. - Renforce la collaboration et la communication au sein de l'équipe. 	<ul style="list-style-type: none"> - Nécessite une forte implication des clients tout au long du projet. - Peut être difficile à mettre en œuvre dans des équipes grandes ou distribuées en raison de l'importance de la communication face à face.
Agile	La méthode agile est une approche de gestion de projet qui favorise la flexibilité face aux changements. Elle vise à développer un produit de qualité de manière incrémentale, contrairement aux méthodes traditionnelles de gestion de projet qui suivent un processus séquentiel rigide.	<ul style="list-style-type: none"> - Flexibilité pour s'adapter aux changements à tout moment du projet. - Fort engagement client, permettant des ajustements fréquents pour mieux répondre aux besoins des utilisateurs 	<ul style="list-style-type: none"> - Moins prévisible en termes de coûts et de délais, ce qui peut être un défi pour la planification et le budget. - Nécessite une communication continue et intensive, ce qui peut être difficile à maintenir sur de longues périodes.
Scrum	Scrum est un cadre Agile conçu pour améliorer la collaboration au sein des équipes, les aidant à accomplir des tâches à haute valeur ajoutée. Il fournit un ensemble de valeurs, de rôles et de règles qui orientent les équipes à se focaliser sur chaque itération et à s'améliorer continuellement	<ul style="list-style-type: none"> - Encourage une livraison rapide et fréquente de fonctionnalités fonctionnelles. Très réactif aux changements, avec des revues régulières du progrès et des obstacles (sprints reviews, daily stand-ups). 	<ul style="list-style-type: none"> - Peut être difficile à adopter pour les équipes habituées aux approches plus traditionnelles. Nécessite une maturité d'équipe et une compréhension des principes Agile pour être efficace.

Table 1.1 – Tableau comparatif des méthodologies de gestion de projet

1.5.2 Choix de la méthodologie Scrum

Nous avons adopté la méthode Scrum pour son efficacité reconnue et sa pertinence pour notre projet. Scrum est un cadre organisationnel pour le développement de projets complexes, offrant une structure flexible basée sur des cycles courts appelés sprints. Chaque sprint, typiquement de deux semaines, commence par une planification et se termine par la livraison d'une fonctionnalité, suivie d'une rétrospective pour améliorer continuellement nos processus grâce à l'auto-organisation de notre équipe.

Les principes de Transparence, Inspection et Adaptation sont essentiels à notre approche :

- **Transparence** : Tous les aspects du processus de développement sont ouverts à tous les participants, assurant que les décisions sont fondées sur une compréhension partagée et complète.
- **Inspection** : Nous évaluons régulièrement nos progrès pour identifier et résoudre les problèmes rapidement, en veillant à ce que ces inspections ne perturbent pas notre productivité.
- **Adaptation** : Répondre de manière proactive aux résultats de nos inspections nous permet d'ajuster nos méthodes et de rester flexibles, essentiel pour s'adapter aux dynamiques de projet changeantes. Cette méthodologie renforce notre capacité à répondre efficacement aux défis, en privilégiant une amélioration continue au cœur de nos opérations.

1.5.3 Processus Scrum

Scrum est une méthode agile de gestion de projet qui intègre des rôles définis et des pratiques structurées pour le développement logiciel. Cette approche, fonctionnant par cycles appelés Sprints de deux à quatre semaines, vise à produire régulièrement des versions fonctionnelles du produit.

1.5.3.1 Processus Scrum Simplifié

1 Planification et Priorisation : Le Product Owner priorise les tâches dans le Product Backlog, équilibrant la valeur et le coût pour répondre aux exigences du marché.

2 Exécution du Sprint : L'équipe planifie et exécute des tâches du Product Backlog pendant le Sprint, avec des check-ins (Daily Scrum) quotidiens pour suivre les progrès.

3 Livraison et Réflexion : À la fin du sprint, les résultats doivent être prêts à être livrés. Avant de commencer un nouveau sprint, l'équipe revient sur les tâches du Product Backlog restantes, répétant le processus jusqu'à ce que le Product Backlog soit épuisé, que le budget soit atteint ou que la date limite soit arrivée.

Ce processus itératif garantit que les fonctionnalités les plus critiques sont toujours prioritaires et que le produit peut évoluer rapidement en réponse aux besoins du client, tout en maintenant une haute qualité grâce à des cycles de développement courts. Scrum est particulièrement apprécié pour sa capacité à faciliter une gestion de projet agile et efficace, adaptée aux environnements dynamiques et aux exigences fluctuantes des projets de développement de logiciels.

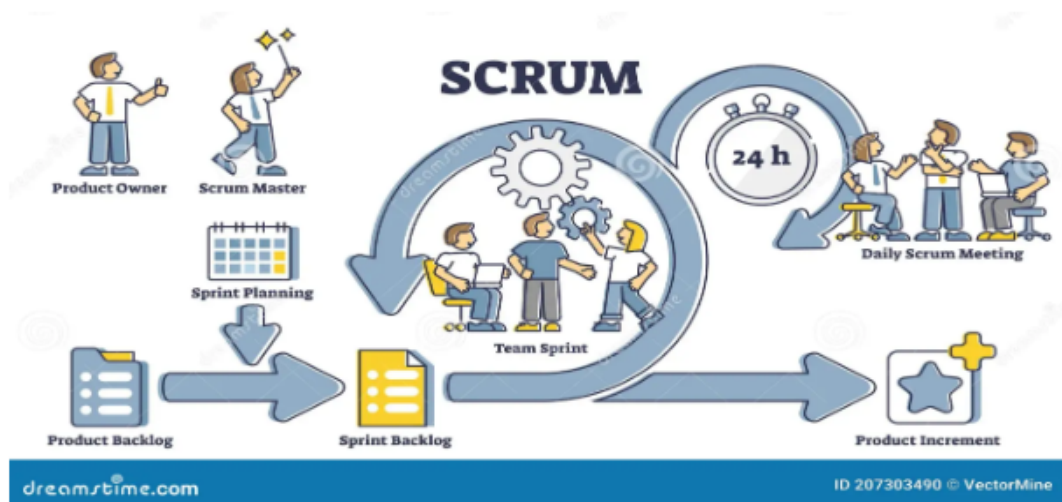


Figure 1.4 – Le diagramme Scrum simplifié

1.6 Langage de modélisation UML

1.6.1 Définition d'UML (Unified Modeling language)

L'UML, ou "Unified Modeling Language", est un langage de modélisation visuelle riche en sémantique et en syntaxe. Il est conçu pour faciliter l'architecture, la conception et la mise en œuvre de systèmes logiciels complexes, en abordant à la fois leur structure et leur comportement. Au-delà du domaine du développement logiciel, UML est également appliqué dans la modélisation des processus industriels. Comparable aux plans utilisés dans d'autres secteurs, UML se compose de divers types de diagrammes. Bien qu'il ne soit pas un langage de programmation, il existe des outils permettant de générer du code en plusieurs langages à partir des diagrammes UML. En général, les diagrammes UML décrivent les limites, la structure et le comportement du système et des objets qu'il contient. Le logo d'UML est présenté par la figure suivante :



Figure 1.5 – Le logo UML

1.6.2 Les principaux objectifs de l'utilisation de l'UML

- Analyser et comprendre le comportement du système.
- Identifier les erreurs et les omissions dès le début du cycle de vie du projet.

-
- Présenter les conceptions proposées et faciliter la communication avec les parties prenantes.
 - Clarifier et documenter les exigences.
 - Guider la mise en œuvre du projet.

1.6.3 Les Différents Diagrammes UML

L'UML propose une variété de diagrammes que les développeurs utilisent pour visualiser les objets, les états et les processus d'un logiciel ou d'un système. Ces diagrammes incluent :

- **Diagramme des cas d'utilisation** : Décrit les interactions entre les utilisateurs et le système.
- **Diagramme de séquence** : Montre l'ordre des interactions entre les objets dans un cas d'utilisation.
- **Diagramme de classe** : Représente les classes du système et leurs relations.
- **Diagramme d'activité** : Visualise le flux de travail ou les activités d'un système.
- **Diagramme d'état** : Modélise les différents états d'un objet et les transitions entre ces états.

1.7 Conclusion

Dans ce chapitre, nous avons exploré le cadre général du projet en introduisant l'organisme d'accueil et le contexte dans lequel s'inscrit ce projet. Nous avons également présenté la problématique. Ensuite, nous avons procédé à une analyse approfondie de l'existant, structurée en plusieurs étapes : une description détaillée de l'état actuel, une critique des aspects existants et la proposition de notre solution. Puis, nous avons discuté de notre approche méthodologique, incluant le choix de la méthode Scrum, une comparaison entre différentes méthodologies et, enfin, une comparaison entre la solution proposée et celle en place.

Chapitre 2 : Planifications et spécifications des besoins

2.1 Introduction

Ce chapitre détaille les besoins fonctionnels et non fonctionnels essentiels à la réalisation d'une application performante, précisant ce qu'elle doit accomplir et comment elle doit fonctionner. Nous identifierons également les acteurs principaux et définirons leur rôle dans le cadre du projet. Ensuite, nous explorerons les diagrammes des cas d'utilisation qui visualisent les interactions entre les acteurs et le système. Nous discuterons également de notre approche méthodologique pour organiser le travail, aboutissant au choix de la méthodologie de gestion de projet. Enfin, nous présenterons l'architecture de l'application, pour illustrer comment les composants du système interagissent et sont intégrés.

2.2 Analyse et spécification des besoins

2.2.1 Les besoins fonctionnelles

- **Vente** : Les agents terrain sont responsables de la vente de puces téléphoniques aux clients, assurant la distribution directe des produits.
- **Pointage** : À chaque ouverture de l'application par un agent terrain, un pointage est enregistré pour suivre l'activité et la présence des agents. .
- **Vérification des ventes** : Ce processus permet de valider ou de refuser les ventes réalisées.
- **Contrôles** : Contrôler les puces vendues pour s'assurer de leur validité et de leur état.

-
- **Gestion des articles** : Permet l'ajout de nouveaux articles dans le système, assurant que l'inventaire reste à jour et précis
 - **Gestion des pertes** : En cas de perte de puces ou d'autres incidents, les agents terrain peuvent déclarer ces événements, permettant un suivi et une gestion appropriés des pertes.
 - **Gestion des transactions** : Notre système enregistre et suit tous les mouvements de stock, éliminant le besoin d'une table de stock séparée. Chaque transaction capte les détails essentiels tels que la source, la destination, l'IccId (ContratNumber), la date d'affectation et le statut de chaque article. Lorsqu'un stock est attribué à un agent, la destination est ajustée pour refléter l'agent spécifique. Ce système assure une gestion précise et transparente des stocks, permettant un suivi complet des entrées, des affectations et des ventes, et contrôle également l'état du stock.
 - **Gestion des utilisateurs** : L'administrateur gère tous les comptes des utilisateurs, y compris ceux des agents, et peut effectuer des modifications ou des mises à jour de leurs profils et de leurs rôles selon les besoins.

2.2.2 Les besoins non fonctionnelles

Après avoir clairement établi les besoins fonctionnels, il est essentiel de considérer les besoins non fonctionnels durant le développement de notre plateforme. Les besoins non fonctionnels se rapportent à des éléments qui, bien que perceptibles par l'utilisateur, ne sont pas directement liés aux fonctionnalités du système. Prendre en compte ces besoins peut améliorer significativement l'expérience utilisateur et le processus de développement du service.

- **Maintenabilité** :

Mises à jour : Le système doit permettre une mise à jour facile et rapide des logiciels, sans temps d'arrêt significatif.

Diagnostic : Des outils de diagnostic doivent être intégrés pour détecter et résoudre rapidement les problèmes.

- **Performance :**

Temps de réponse : Le système doit répondre aux requêtes des utilisateurs en moins de 2 secondes sous charge normale.

Capacité : Le système doit être capable de gérer simultanément jusqu'à 1000 utilisateurs actifs sans dégradation des performances.

- **Sécurité :**

Authentification : Tous les utilisateurs doivent être authentifiés avant d'accéder à leurs fonctionnalités spécifiques.

Autorisation : Les droits d'accès doivent être strictement définis et contrôlés selon les rôles des utilisateurs.

- **Utilisabilité :**

Interface utilisateur : L'interface doit être intuitive et facile à utiliser pour tous les types d'utilisateurs, minimisant le besoin de formation.

- **Compatibilité :**

Intégration : Le système doit être capable de s'intégrer facilement avec d'autres systèmes ou logiciels existants.

Support multiplateforme : L'application doit fonctionner de manière fluide sur divers systèmes d'exploitation tels que Windows, MacOS, Android et Ios.

2.3 Identification et structuration de cas d'utilisation

2.3.1 Identification des acteurs du système

Un acteur définit le rôle joué par une personne, un processus ou une entité qui interagit directement avec le système. Pour notre application, les acteurs identifiés sont les suivants :

Rôle	Descriptions
Administrateur	Responsable de la gestion globale de l'application, cet employé interagit directement avec le back-office pour administrer les fonctionnalités du système.
Agent Terrain	L'Agent de Terrain opère directement sur le terrain pour la vente et la distribution des produits. Ses principales responsabilités comprennent non seulement la gestion des transactions commerciales avec les clients, mais aussi cet agent a une tâche cruciale de pointage : au début de sa journée de travail, il doit enregistrer sa présence et sa localisation via l'application.
Agent saisie	Chargé de traiter et valider les transactions de vente et de pointage, cet acteur joue un rôle crucial dans la vérification de la précision des données de vente enregistrées.
Agent stock	Gère le stock et assure le suivi des inventaires. Il est responsable de la réception des nouveaux stocks et de l'ajout des nouveaux articles.

Table 2.1 – Les acteurs et leurs descriptions

2.3.2 Diagramme des Cas d'utilisations global

Le Diagramme de cas d'utilisation est utilisé pour représenter les fonctionnalités d'un système et définir ses spécifications. Il expose les fonctionnalités clés et le domaine d'application du système, ainsi que les échanges entre le système et ses acteurs. Ce diagramme illustre le comportement dynamique d'un système en fonctionnement et illustrent les entités externes qui interagissent avec ce système.



2.4 Pilotage du projet avec scrum

Cette section se compose de trois éléments cruciaux du Scrum, à savoir les rôles, le backlog du projet, et la planification.

2.4.1 Equipe Et Rôle

- **Product Owner** : Responsable de maximiser la valeur du produit en gérant le Product Backlog. Le Product Owner priorise les items du backlog en fonction de leur valeur pour le client ou l'utilisateur final.
- **Scrum Master** : Facilite le processus Scrum pour l'équipe de développement et le Product Owner. Le Scrum Master aide à résoudre les obstacles, et assure une dynamique de travail fluide.
- **Scrum Team** : Un groupe auto-organisé qui réalise le travail technique nécessaire pour développer les incréments du produit.

Rôle Scrum	Personnes Affectées
Product Owner	M. Marwen Khouaja
Scrum Master	M. Marwen Khouaja
Scrum Team	Kamel Jemmali Yassine Chouchene

Table 2.2 – Rôles Scrum et personnes affectées

2.4.2 Backlog du produit

Représente une liste évolutive de toutes les tâches à exécuter, sous la supervision du Product Owner. Cette liste flexible inclut des fonctionnalités, des spécifications, des améliorations et des correctifs, et sert de fondement pour le backlog de sprint, agissant comme la liste de tâches principale pour les équipes. Le Product Owner réajuste continuellement et réorganise les priorités du backlog en fonction des évolutions du marché ou des nouvelles découvertes, adaptant ou modifiant les tâches selon les nécessités émergentes.

ID	Histoires	Durée (jours)	Priorité
1	En tant qu'administrateur, je peux m'authentifier	2	1
2	En tant qu'administrateur, j'ai la possibilité de valider une vente	5	1
3	En tant qu'administrateur, j'ai la possibilité de refuser une vente	2	1
4	En tant qu'administrateur, j'ai la possibilité de consulter tous les ventes	3	1
5	En tant qu'administrateur, j'ai la possibilité d'ajouter un article	1	1
6	En tant qu'administrateur, j'ai la possibilité de supprimer un article	1	3
7	En tant qu'administrateur, j'ai la possibilité de modifier un article	1	3
8	En tant qu'administrateur, j'ai la possibilité de consulter tous les articles	1	3
9	En tant qu'administrateur, j'ai la possibilité d'affecter le stock	6	1
10	En tant qu'administrateur, j'ai la possibilité d'entrer stock	4	1
11	En tant qu'administrateur, j'ai la possibilité de consulter l'état du stock terrain et saisie	3	2
12	En tant qu'administrateur, j'ai la possibilité de créer des comptes	2	1
13	En tant qu'administrateur, j'ai la possibilité de supprimer des comptes	1	2
14	En tant qu'administrateur, j'ai la possibilité de modifier des comptes	1	2
15	En tant qu'administrateur, j'ai la possibilité de consulter tous les comptes	1	3
16	En tant qu'agent saisie, je peux m'authentifier	2	1
17	En tant qu'agent saisie, j'ai la possibilité de contrôler les pointages	1	3
18	En tant qu'agent saisie, j'ai la possibilité de contrôler les ventes	2	1
19	En tant qu'agent saisie, j'ai la possibilité de valider une vente	5	1
20	En tant qu'agent saisie, j'ai la possibilité de refuser une vente	2	1
21	En tant qu'agent saisie, j'ai la possibilité de télécharger la liste des pointages et des ventes (pdf-excel)	2	3
22	En tant qu'agent stock, je peux m'authentifier	2	1
23	En tant qu'agent stock, j'ai la possibilité d'ajouter un article	1	1
24	En tant qu'agent stock, j'ai la possibilité de modifier un article	1	2
25	En tant qu'agent stock, j'ai la possibilité de supprimer un article	1	2
26	En tant qu'agent stock, j'ai la possibilité d'afficher tous les articles	1	2
27	En tant qu'agent stock, j'ai la possibilité d'entrer le stock	4	1
28	En tant qu'agent stock, j'ai la possibilité d'affecter le stock	6	1

29	En tant qu'agent stock, j'ai la possibilité de consulter l'état du stock terrain et saisie	3	2
30	Mobile : En tant qu'agent terrain, je peux m'authentifier	3	1
31	Mobile : En tant qu'agent terrain, je peux pointer	8	1
32	Mobile : En tant qu'agent terrain, je peux ajouter une vente	5	1
33	Mobile : En tant qu'agent terrain, je peux réclamer une perte	0	3
34	Mobile : Afficher les ventes du jour	2	2
34	Mobile : Consulter mon compte	1	3

Table 2.3 – Le backlog Produit

2.4.3 Planification des sprints

Cette session implique le Product Owner et l'équipe de développement qui déterminent les activités du prochain sprint. Le résultat est le Sprint Backlog, qui consiste en une sélection d'éléments du Product Backlog assortie d'un plan d'action pour leur mise en œuvre.

Nous avons identifié quatre sprints :

Sprint 1 : «Agent Terrain et Superviseur»

Sprint 2 : «Agent Saisie»

Sprint 3 : «Agent Stock»

Sprint 4 : «Administrateur»

2.5 Environnement de travail

2.5.1 Environnement Matériel

Dans cette section, nous détaillons les ressources matérielles employées pour mener à bien ce projet. Au cours des différentes étapes du projet, nous disposons de deux ordinateurs portables, dotés des caractéristiques suivantes :

Marque	Processeur	RAM	S.E
Asus Rog	Intel(R) Core(TM) i7 7ème génération	12,0 Go	Windows 11
Asus Vivobook	Intel(R) Core(TM) i5 10ème génération	16,0 Go	Windows 11

Table 2.4 – Environnement Matériel

2.5.2 Environnement Logiciel

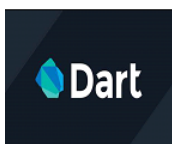
Dans cette section, nous détaillons les ressources Logiciels employées pour mener à bien ce projet.

Langages et Technologies



Flutter

Flutter est un framework de développement open-source conçu par Google pour la création d'applications mobiles multiplateformes. Il permet de développer simultanément pour iOS et Android avec un seul code source, utilisant le langage Dart, également de Google. Ce framework se distingue par son moteur graphique avancé qui rend l'interface utilisateur directement, offrant ainsi des animations lisses et des designs attractifs.



Dart

Dart est un langage de programmation créé par Google, conçu pour une utilisation polyvalente dans divers domaines tels que le développement d'applications mobiles, le développement web, et la création de scripts côté serveur.



Flask

Flask est un microframework web léger et flexible pour Python. Conçu pour être simple et minimaliste, Flask fournit les éléments essentiels pour créer des applications web, tout en permettant aux développeurs de choisir et d'ajouter les extensions selon leurs besoins.



Python

Python est un langage de programmation polyvalent et de haut niveau, connu pour sa simplicité et sa clarté. Conçu pour être facile à apprendre et à utiliser, Python offre une syntaxe concise qui améliore la productivité des développeurs. Avec sa vaste bibliothèque standard et ses nombreux modules et frameworks, Python est adapté à diverses applications, allant du développement web et scientifique à l'intelligence artificielle et à l'automatisation des tâches.



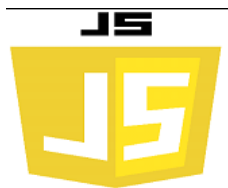
ASP.NET Core MVC

ASP.NET Core MVC est un framework qui facilite la création d'applications web et d'APIs en utilisant le modèle de conception Model-View-Controller.



C#

C# est un langage de programmation fortement typé et orienté objet, développé par Microsoft. Il est couramment utilisé pour créer des applications web, des services web, des applications de bureau, des scripts et des widgets principalement sur la plateforme .NET.



JavaScript

JavaScript est un langage de programmation conçu pour rendre les sites web interactifs et dynamiques. Il s'agit d'un langage de script qui peut fonctionner à la fois dans le navigateur de l'utilisateur et sur un serveur.



HTML5

HTML5 est une version avancée du langage de balisage HTML, qui est reconnu comme le standard pour la création de pages web.



Le CSS

Le CSS, ou Cascading Style Sheets (feuilles de style en cascade en français), est un langage informatique utilisé pour styliser et mettre en forme les documents HTML sur Internet. Développé dans les années 1990, le CSS offre une alternative à la mise en forme directe via des balises HTML, permettant une séparation plus claire entre le contenu et la présentation visuelle.

Outils de développement



Visual Studio Code (VS Code)

Visual Studio Code (VS Code) est un éditeur de code source gratuit développé par Microsoft. Il supporte de nombreux langages de programmation tels que JavaScript, Python, et C++, et est compatible avec Windows, macOS, et Linux. VS Code est apprécié pour son interface épurée et sa capacité à intégrer des extensions qui augmentent sa fonctionnalité, rendant le développement plus efficace et personnalisable.



Microsoft Visual Studio

Microsoft Visual Studio est un environnement de développement intégré (EDI) conçu par Microsoft pour la création de logiciels, sites web et applications mobiles. Il prend en charge divers langages de programmation tels que C, C++, et Visual Basic .NET, et propose des fonctionnalités pour l'édition de code, le débogage et la gestion des versions. Visual Studio optimise l'efficacité des développeurs et s'intègre avec des services cloud comme Azure.

Outil et Framework de Tests



Postman

Postman est un outil utilisé pour tester des services web, permettant aux développeurs de vérifier et de déboguer des API.

Outils de contrôle de versions du code source



Git Git est un système de contrôle de version distribué, très populaire et gratuit, conçu pour gérer efficacement les projets de toute taille avec rapidité et précision. Git est particulièrement puissant pour coordonner le travail parmi les programmeurs collaborant sur le code source lors du développement logiciel.



GitHub GitHub est une plateforme d'hébergement pour le développement logiciel, basée sur le système de gestion de versions Git. Elle offre aux développeurs la possibilité de stocker et de partager leurs projets Git sur le web.

Outil de conceptions



Draw.io Draw.io est un outil gratuit et open-source pour créer des diagrammes en ligne. Il supporte de nombreux types de diagrammes comme les diagrammes UML (classe, séquence, cas d'utilisation).

Serveurs de base de données



SQL Server SQL Server est un système de gestion de base de données (SGBD) conçu et commercialisé par Microsoft. Ce produit est un élément clé de la stratégie technique de Microsoft pour les solutions de gestion de données.

2.6 Architecture de l'application

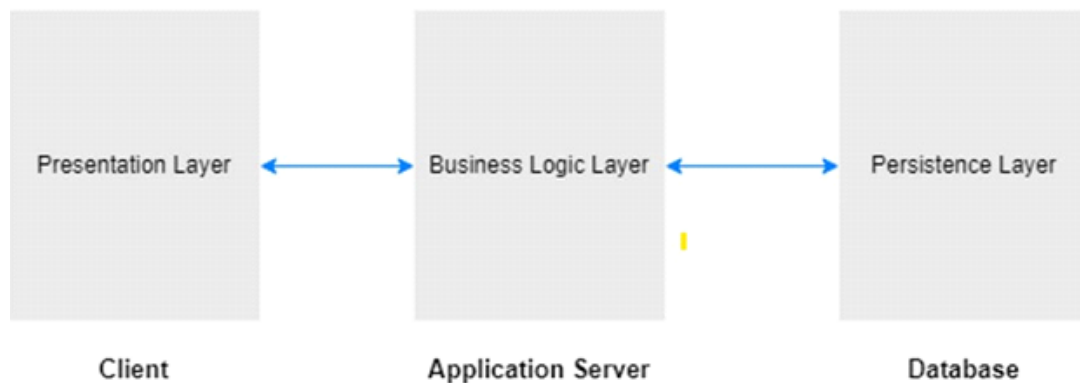
2.6.1 Architecture logique

L'architecture choisie pour la solution est de type trois tiers. Ce modèle d'architecture logique se divise en trois couches distinctes, chacune avec ses propres responsabilités, au sein d'un système unique.

- **couche de présentation** : Cette couche constitue l'interface utilisateur et le canal de communication de l'application. Elle permet à l'utilisateur final d'interagir avec l'application. Son rôle principal est d'afficher des informations aux utilisateurs et de collecter les don-

nées qu'ils saisissent. Le Product Owner priorise les items du backlog en fonction de leur valeur pour le client ou l'utilisateur final.

- **La couche logique** : Également appelée couche d'application ou couche intermédiaire, elle traite les informations recueillies par la couche de présentation en appliquant la logique métier. Elle peut aussi supprimer, ajouter ou modifier des données dans la couche de données.
- **La couche de données** : Aussi connue sous les noms de couche de base de données, ou couche de persistance, cette couche est responsable du stockage et de la gestion des informations utilisées par l'application.



2.6.2 Architecture Logicielle

- **Modèle** : Cette composante gère les données utilisées par l'application. Elle récupère les données de la base de données, les organise, et les prépare pour être traitées par le contrôleur.
- **Vue** : La vue est l'élément graphique de l'interface utilisateur qui présente les données du modèle à l'utilisateur. Elle peut être créée à l'aide de langages de balisage tels que CSS, JavaScript et HTML.

- **Contrôleur** : Le contrôleur est responsable de la gestion des interactions de l'utilisateur avec l'application web. Il reçoit les requêtes HTTP du navigateur et sert d'intermédiaire entre la vue et le modèle. Le contrôleur obtient les données du modèle, les traite et les envoie à la vue pour qu'elles soient affichées. Il gère également les actions des utilisateurs, telles que la soumission d'un formulaire, et met à jour le modèle en conséquence.

L'architecture MVC permet de séparer distinctement la logique métier, la présentation et les interactions utilisateur. Cette séparation favorise une meilleure modularité et maintenabilité du code. Par exemple, pour changer l'apparence de votre application web, vous pouvez modifier la vue sans affecter le modèle ou le contrôleur. De même, les modifications de la logique métier peuvent être effectuées dans le modèle sans impacter la vue.

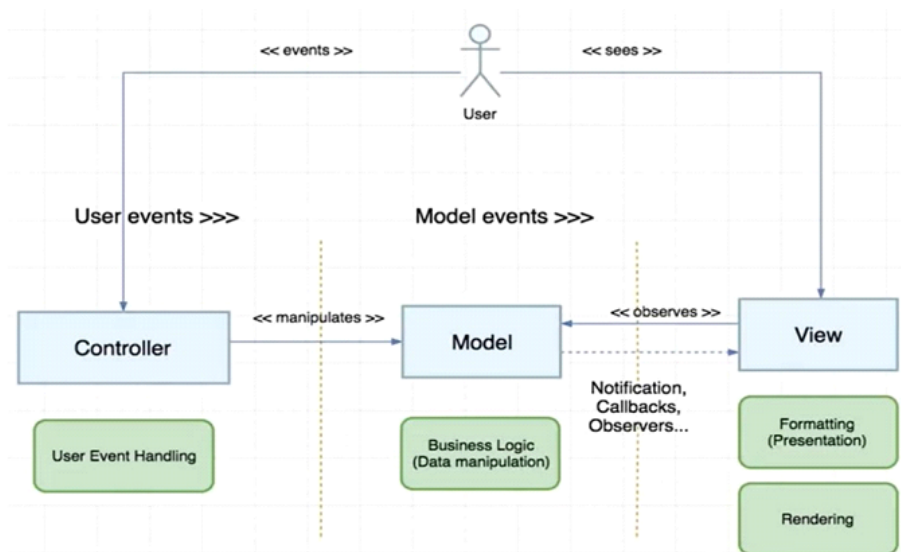


Figure 2.2 – MVC WEB

2.7 Conclusion

Au cours de ce chapitre, nous avons détaillé les besoins fonctionnels et non fonctionnels qui guident le développement de notre application, identifié les principaux acteurs impliqués et clarifié leurs rôles. Un diagramme de cas d'utilisation a été présenté pour illustrer les interactions système-utilisateur. Nous avons décrit la gestion du backlog produit, et la planification des sprints. L'environnement matériel et logiciel nécessaire au projet a également été examiné en détail. Le chapitre suivant entamera l'étude approfondie du premier sprint, marquant ainsi le début de l'implémentation pratique de notre projet.

Chapitre 3 : Etude et réalisation du sprint1

3.1 Introduction

Ce chapitre va présenter le premier sprint « Application Mobile », en détaillant les tâches essentielles associées à chaque User Story. Ces tâches seront accompagnées de diagrammes UML qui illustreront la conception et la logique appliquées au cours des différentes itérations.

3.2 Sprint 1 : « Application Mobile »

Les agents de terrain sont au cœur des opérations de vente et de service client. Ce chapitre décrit les fonctionnalités de l'application mobile utilisées par ces agents pour réaliser des ventes, enregistrer des pointages, et déclarer des pertes. Nous examinerons comment l'application permet aux agents de terrain de travailler efficacement sur le terrain, grâce à des outils comme la géolocalisation et la reconnaissance faciale. Le rôle des agents de terrain est vital pour garantir une interaction fluide et productive avec les clients, en assurant des services rapides et fiables.

3.2.1 Backlog du Sprint 1

Le tableau ci-dessous donne un aperçu détaillé sur le Backlog du quatrième sprint qui prend en charge plusieurs fonctionnalités d'agent terrain et superviseur. **«Application Mobile»**

ID	Histoire	Estimation/jours
1	En tant qu'agent Terrain , je peux s'authentifier	3
2	En tant qu'agent Terrain , j'ai la possibilité d'ajouter une vente	5
3	En tant qu'agent Terrain , j'ai la possibilité d'afficher mes ventes du jour	2
4	En tant qu'agent Terrain , j'ai la possibilité de réclamer à une perte	0
5	En tant qu'agent Terrain , j'ai la possibilité de pointer	8
6	En tant qu'agent Terrain , j'ai la possibilité de consulter mon compte	1

Table 3.1 – Tableau des histoires et estimations en jours du sprint 1

3.2.2 Spécification fonctionnelle du Sprint 1

Cette section introduit la phase d'analyse, centrée sur la fonctionnalité de l'agent de Terrain en répondant à la question : "Que fait l'agent de Terrain?" Les réponses sont illustrées à travers un diagramme de cas d'utilisation, complété par des descriptions textuelles détaillées pour chaque cas d'utilisation.

3.2.2.1 Diagramme de cas d'utilisation du Sprint 1

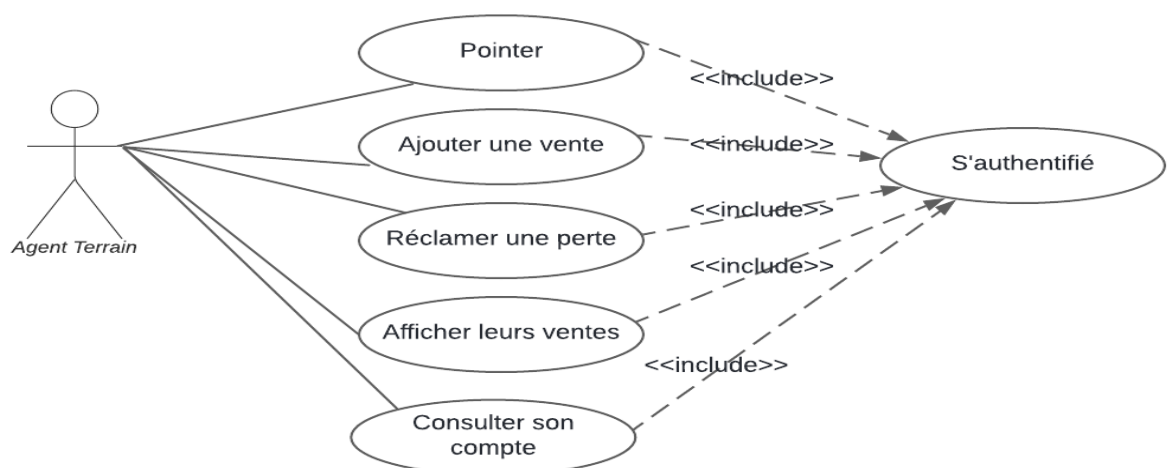


Figure 3.1 – Diagramme de cas d'utilisation pour sprint 1

3.2.2.2 Description textuelle de cas d'utilisations

«Pointer»

Titre	Pointage
Acteur	Agent de Terrain
Précondition	Agent de Terrain authentifié avec succès
Scénario Nominal	1. L'application demande l'autorisation d'accéder à la localisation de l'appareil 2. L'application hérite le nom, le prénom et l'IdUserTerrain de la page de login. 3. L'Agent de Terrain prend une photo pour le pointage.
Post Condition	L'Agent de Terrain accède le menu d'AddVente.

Table 3.2 – Description textuelle de cas d'utilisation «Pointer»

« Ajouter une Vente »

Titre	Ajouter une Vente
Acteur	Agent de Terrain
Précondition	Agent de Terrain fait son pointage avec succès
Scénario Nominal	1. L'Agent de Terrain remplit le formulaire avec une vérification de tous les champs. 2. L'application hérite l'IdUserTerrain de la page de pointage. 3. L'Agent de Terrain ajoute les images de CIN et du contrat 4. Le système renvoie ensuite à la liste mise à jour des ventes.
Post Condition	La vente est ajouté a la liste des ventes.

Table 3.3 – Description textuelle de cas d'utilisation « Ajouter une Vente »

3.2.2.3 Diagramme de classe du Sprint 1

Le diagramme de classes est un outil permettant de représenter la structure interne d'un système en exposant les différentes classes, leurs attributs, ainsi que les relations structurelles qui les lient.

La figure ci-dessous illustre le diagramme de classes que nous avons utilisé pour développer le premier sprint.

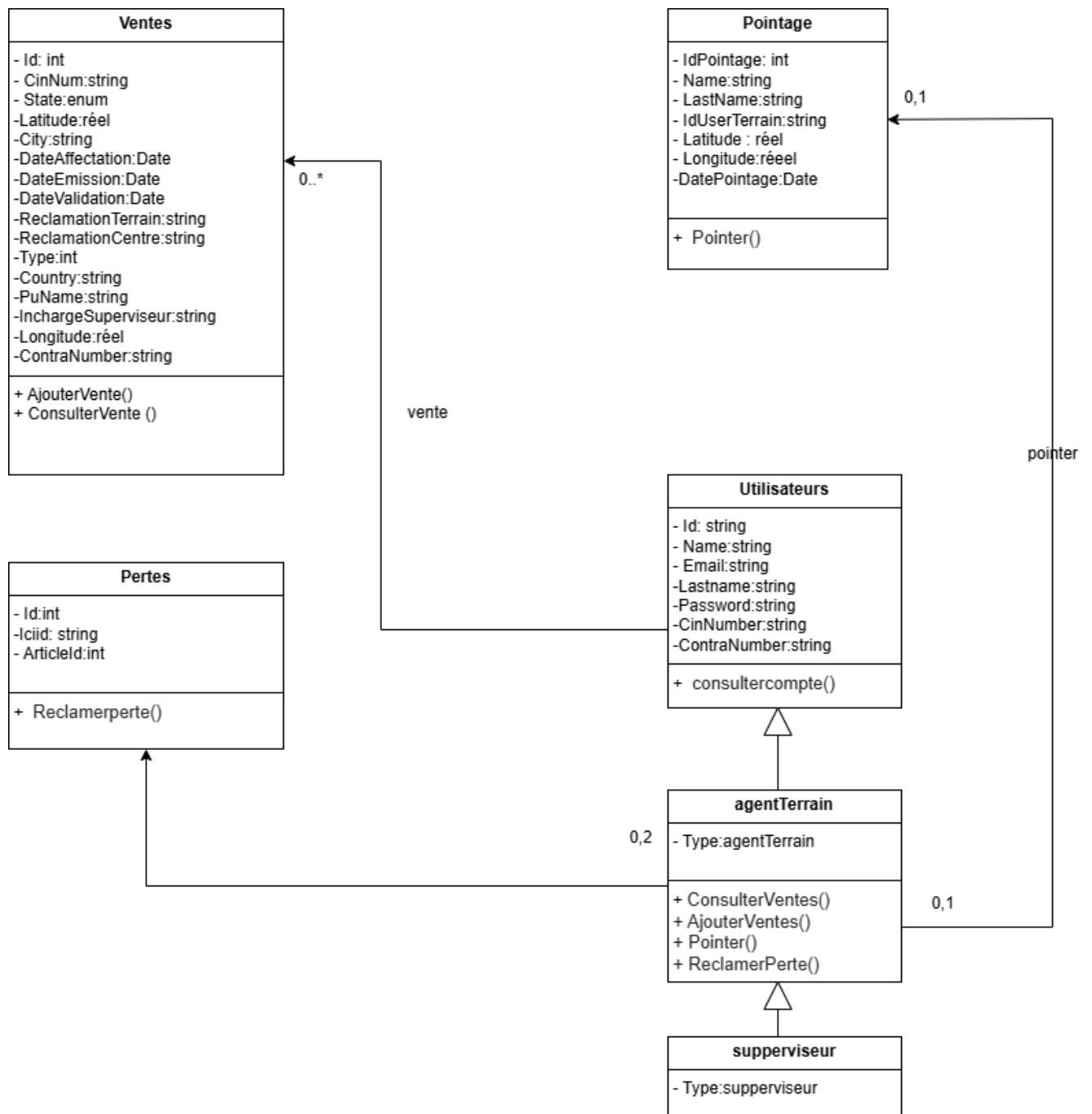


Figure 3.2 – Diagramme de classe pour le sprint 1

3.2.2.4 Diagramme de séquence

Afin de représenter les communications internes de notre système informatique et avec les acteurs externes, nous employons le diagramme de séquence UML. Ce diagramme permet de visualiser de manière chronologique les interactions entre les objets et les acteurs, ainsi que les messages échangés entre eux.

- **Diagramme de séquence détaillée(login)**

Ce diagramme de séquence illustre le processus d'authentification pour un utilisateur dans une application mobile. Il se divise en deux scénarios principaux : un utilisateur déjà authentifié aujourd'hui et un utilisateur qui ne l'est pas encore. Si l'utilisateur est déjà authentifié et a pointé, l'application affiche l'interface de vente. Si l'utilisateur est authentifié mais n'a pas encore pointé, l'interface de pointage est affichée. Pour les utilisateurs non authentifiés, l'application présente d'abord l'interface de login. Après que l'utilisateur saisit son nom d'utilisateur et son mot de passe, les données sont vérifiées et envoyées pour rechercher l'utilisateur et déterminer son type. Si l'utilisateur est trouvé et est de type "agent terrain", l'interface de pointage est affichée. Sinon, si l'utilisateur n'est pas trouvé ou s'il est d'un type différent, un message de login incorrect est affiché. Ce diagramme montre les différentes interactions entre les composants de l'application (utilisateur, vue de login, contrôleurs, et modèle utilisateur) pour gérer le processus d'authentification.

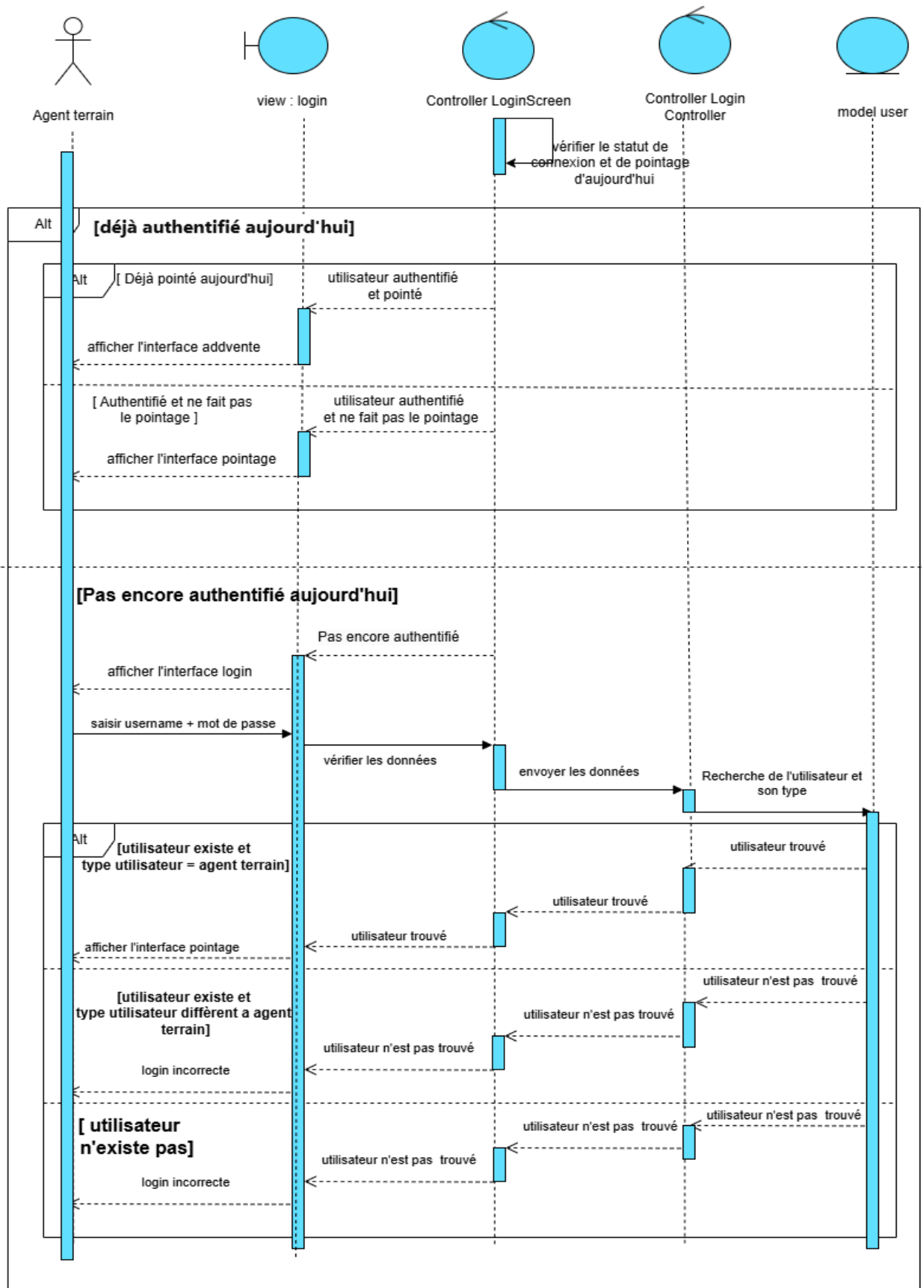


Figure 3.3 – Diagramme de séquence « login du sprint 1 »

○ Diagramme de séquence détaillée(Ajouter une vente)

Ce diagramme de séquence UML illustre le processus d'ajouter une vente par un Agent de Terrain, soulignant les interactions entre divers composants du système : Le processus commence par la saisie des données de vente dans l'interface utilisateur dédiée. Ces données sont transmises au contrôleur de vente, qui orchestre la sauvegarde des informations dans la base de données. En parallèle, les images associées à la vente sont téléchargées et stockées dans un système de gestion des fichiers. Si l'opération est réussie, un message de succès est renvoyé et l'Agent de Terrain est redirigé vers la page affichant toutes les ventes. En cas d'erreur, un message approprié est affiché et l'agent peut être redirigé vers la page d'ajout pour corriger les données.

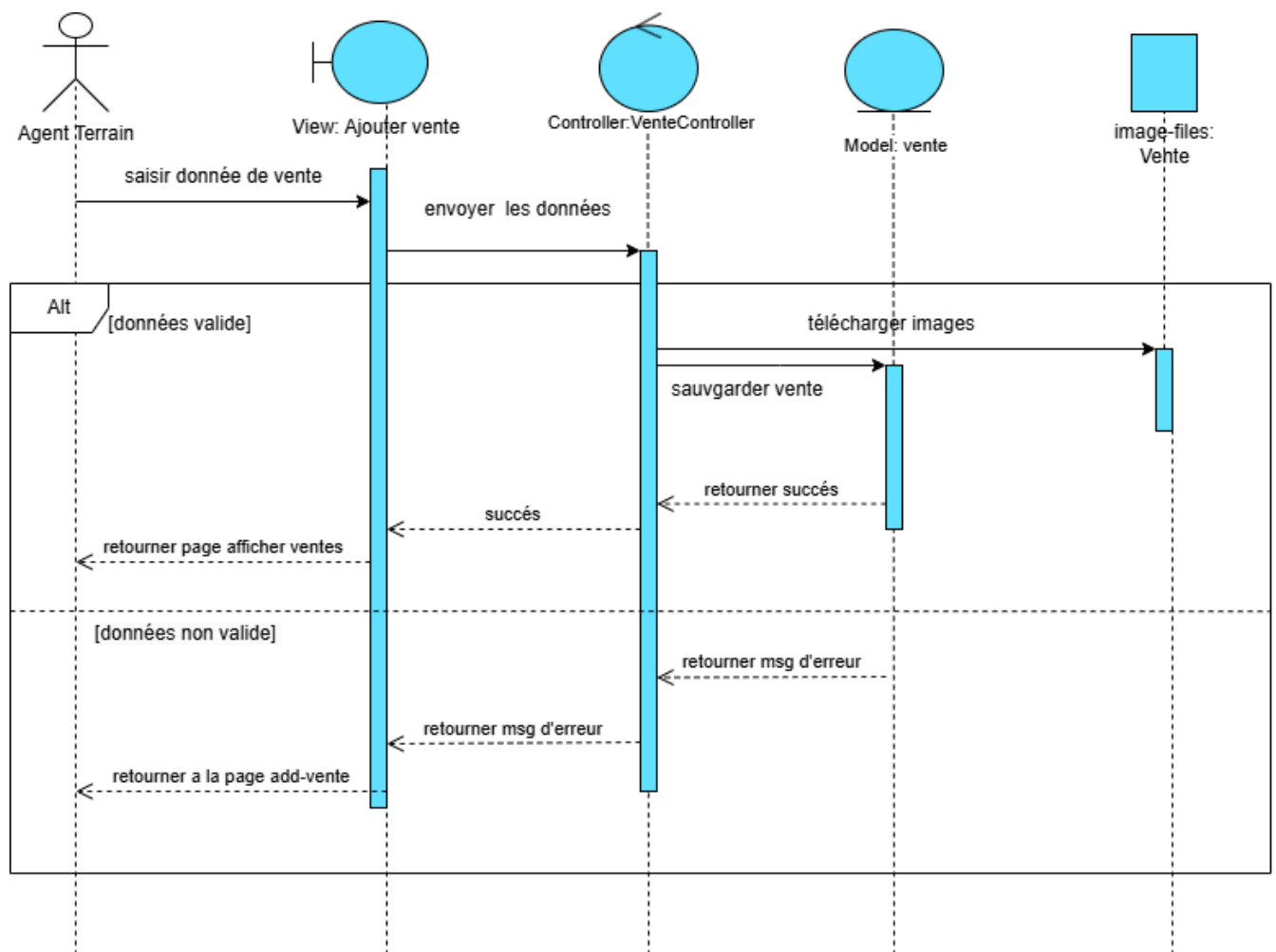


Figure 3.4 – Diagramme de séquence « Ajouter une Vente »

- **Diagramme de séquence détaillée(Pointage)**

Ce diagramme de séquence UML illustre le processus de pointage par un agent de terrain, en soulignant les interactions entre les différents composants du système. Le processus commence par l'authentification de l'agent de terrain. Ensuite, l'application demande la permission d'activer la localisation. Si la localisation est activée, l'agent de terrain doit prendre une photo pour le pointage. Si l'agent ne prend pas de photo ou si l'image prise ne correspond pas à celle enregistrée dans le compte de l'agent, le système renvoie un message d'erreur. En revanche, si les images sont compatibles, le système affiche l'interface permettant d'ajouter une vente. Si l'agent de terrain ne permet pas l'activation de la localisation, le système redemande la permission d'activer la localisation.

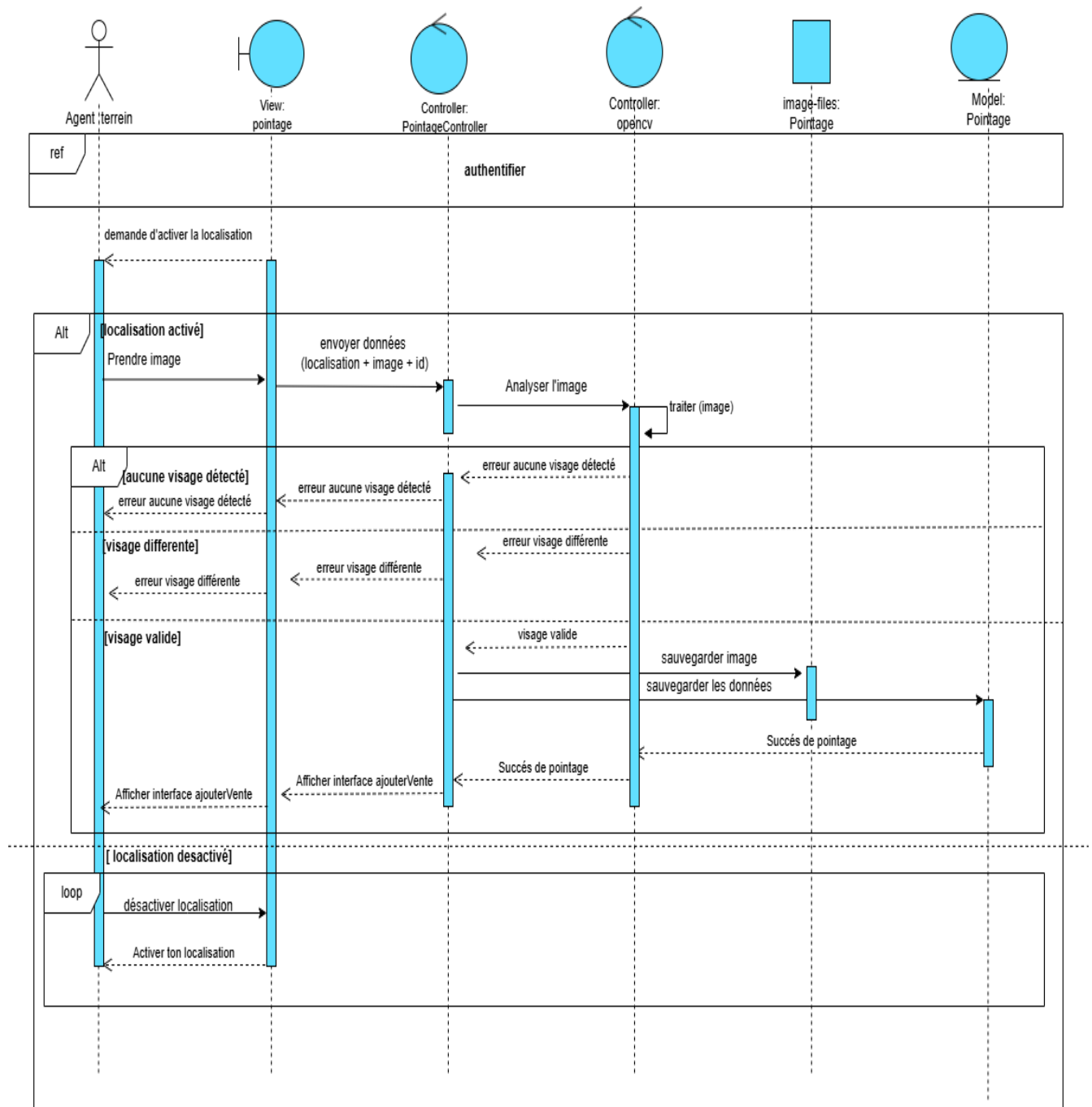


Figure 3.5 – Diagramme de sequence « Pointer »

3.2.3 SignalR

Notre système est confronté à un défi crucial : la nécessité d'une communication en temps réel pour la transmission efficace des ventes aux agents de saisie. Dès qu'une vente est effectuée, il est impératif que l'information parvienne instantanément à l'agent concerné. Cette dynamique

de communication a été abordée dans la section précédente. Techniquement, nous avons surmonté ce défi en intégrant la bibliothèque "SignalR" développée par Microsoft. SignalR facilite la communication en temps réel entre le client et le serveur. Elle offre une API simplifiée pour établir des appels de procédures à distance (RPC) du serveur vers le client, permettant d'exécuter des fonctions JavaScript dans les navigateurs des utilisateurs directement depuis le code .NET côté serveur. En utilisant SignalR, notre système assure que chaque vente est immédiatement visible pour l'agent de saisie, renforçant ainsi la réactivité et l'efficacité de notre processus de vente.

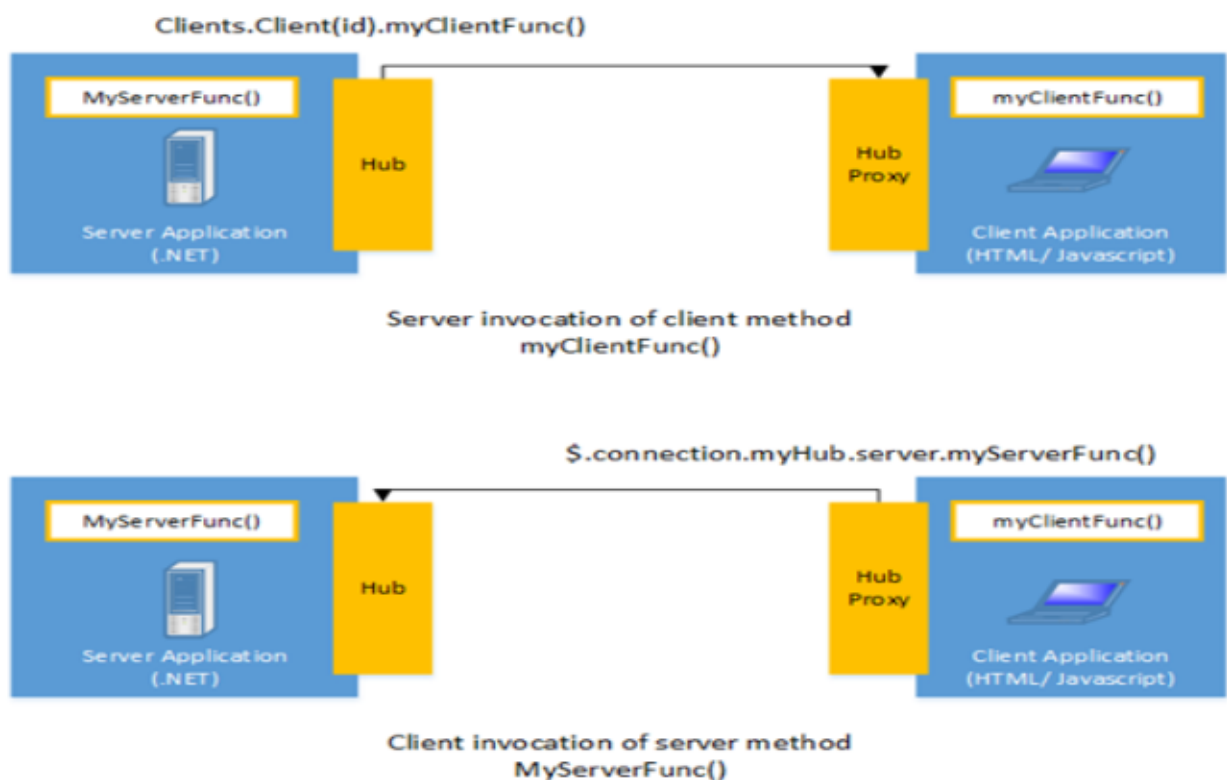


Figure 3.6 – Appels De Procédures à Distance En SignalR

3.2.3.1 Fonctionnement de SignalR

SignalR tire parti des WebSockets pour offrir une communication bidirectionnelle en temps réel entre clients et serveurs. Voici comment cela fonctionne :

1-Connexions Persistantes :

Contrairement aux requêtes HTTP traditionnelles, qui sont sans état, SignalR crée et main-

tient des connexions persistantes. Cela permet une interaction continue et en temps réel.

2-Choix du Transport : SignalR sélectionne automatiquement le meilleur moyen de transport en fonction des capacités du client et de la configuration du serveur. Les méthodes de transport incluent :

- **WebSockets** : C'est l'option la plus efficace quand elle est supportée par le serveur et le client.
- **Server-Sent Events** : Utilisé pour les clients qui supportent HTML5 lorsque WebSockets n'est pas disponible.
- **Long Polling** : Une méthode de repli où SignalR envoie périodiquement des requêtes au serveur et attend une réponse avant d'en émettre une nouvelle, utilisée quand les autres options ne sont pas possibles.

3-Hubs : Les hubs offrent un cadre de communication de haut niveau qui facilite les appels de méthodes bidirectionnels entre clients et serveurs. Ils jouent un rôle crucial dans la gestion des connexions et l'orchestration des interactions client-serveur.

3.2.4 Reconnaissance Faciale

Notre système de pointage doit relever un défi majeur : assurer la reconnaissance faciale en temps réel pour vérifier l'identité des employés lors de chaque pointage. Dès qu'un employé effectue un pointage, il est crucial que l'image de son visage soit immédiatement comparée aux données stockées pour valider son identité. Nous avons résolu ce défi technique en intégrant une API Python spécialisée



Figure 3.7 – Reconnaissance faciale

dans la reconnaissance faciale, qui communique avec notre application ASP.NET via une inter-

face RESTful. Cette intégration permet d'exploiter les avancées de l'IA et du machine learning en Python tout en maintenant une communication transparente avec notre application principale. Chaque image capturée lors du pointage est rapidement traitée par l'API Python, et les résultats de la reconnaissance faciale sont instantanément renvoyés à l'application .NET. Par conséquent, notre système garantit que chaque pointage est vérifié sans délai, améliorant ainsi la sécurité et l'efficacité de notre processus de gestion des présences.

3.2.4.1 Les Relations entre l'Intelligence Artificielle, la machine learning, le deep learning et la data science

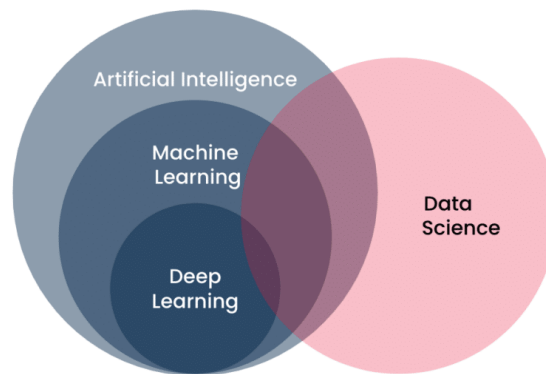


Figure 3.8 – ai structure

L'image représente un diagramme de Venn illustrant les relations entre quatre domaines interconnectés : l'intelligence artificielle (IA), l'apprentissage automatique (Machine Learning), l'apprentissage profond (Deep Learning) et la science des données (Data Science).

- **Science des Données (Data Science)** : La science des données est un domaine interdisciplinaire qui utilise des méthodes, des processus, des algorithmes et des systèmes scientifiques pour extraire des connaissances et des insights à partir de données structurées et non structurées. Ce domaine utilise des techniques d'IA, de machine learning et de deep learning pour analyser et interpréter les données.

-
- **Intelligence Artificielle (IA)** : L'IA est un domaine vaste qui vise à créer des systèmes capables de simuler l'intelligence humaine. Cela inclut des tâches telles que la reconnaissance vocale, la prise de décision, la compréhension du langage naturel et la vision par ordinateur. Le cercle rose le plus grand englobe tous ces aspects.
 - **Apprentissage Automatique (Machine Learning)** : Le Machine Learning est une sous-discipline de l'IA. Il se concentre sur le développement d'algorithmes qui permettent aux machines d'apprendre à partir des données. Plutôt que de suivre des instructions programmées de manière explicite, les systèmes de machine learning améliorent leurs performances en analysant des exemples et en identifiant des modèles.
 - **Deep Learning** : Le deep learning, ou apprentissage profond, est une sous-branche du machine learning qui utilise des réseaux de neurones artificiels avec plusieurs couches cachées (profondeur). Ces réseaux complexes sont capables d'apprendre des représentations de données à différents niveaux d'abstraction, ce qui leur permet de résoudre des tâches complexes de manière plus efficace que les algorithmes traditionnels de machine learning. Voici quelques algorithmes couramment utilisés en deep learning :
 - **Réseaux de Neurones Récurrents (RNN)** :

Adaptés pour les données séquentielles, comme les séries temporelles et le traitement du langage naturel.
 - **ANN (Artificial Neural Network)** :

Les Artificial Neural Networks (ANNs), ou réseaux de neurones artificiels, sont une classe plus générale de réseaux de neurones qui imitent la structure et le fonctionnement du cerveau humain. Les ANNs sont constitués de couches de neurones artificiels (ou nœuds) connectés entre eux. Chaque connexion a un poids ajustable qui est appris lors de l'entraînement.
-

Les ANNs sont utilisés dans une grande variété de domaines, y compris la reconnaissance vocale, la traduction automatique, les systèmes de recommandation, la détection de fraudes, et plus encore.

- **CNN (Convolutional Neural Network) :**

Les Convolutional Neural Networks (CNNs) sont un type de réseau de neurones spécialement conçu pour traiter les données structurées en grille, comme les images. Les CNNs utilisent des opérations de convolution pour extraire des caractéristiques (features) locales de l'entrée, réduisant ainsi la complexité et la quantité de paramètres par rapport aux réseaux de neurones traditionnels. Voici quelques caractéristiques clés des CNNs :

- ◇ **Couches de Convolution** : Ces couches appliquent des filtres (ou noyaux) à l'image d'entrée pour extraire des caractéristiques locales.
- ◇ **Couches de Pooling** : Ces couches réduisent la dimensionnalité des caractéristiques extraites, ce qui permet de diminuer le nombre de paramètres et de contrôler le surapprentissage.
- ◇ **Couches Fully Connected (FC)** : Après les couches de convolution et de pooling, une ou plusieurs couches pleinement connectées sont utilisées pour la classification finale.

Les CNNs sont principalement utilisés pour des tâches de vision par ordinateur telles que la reconnaissance d'images, la détection d'objets, la segmentation d'images et la reconnaissance de visages.

→ Les réseaux de neurones convolutifs (CNN) sont préférés aux réseaux de neurones artificiels (ANN) pour les tâches de vision par ordinateur en raison de leur capacité à extraire des caractéristiques locales et à gérer les variations spatiales dans les images. Les CNN utilisent des filtres partagés pour réduire le nombre de paramètres et les couches de pooling pour diminuer la

dimensionnalité des données, ce qui les rend plus efficaces et moins gourmands en ressources. En revanche, les ANN traitent chaque pixel indépendamment et nécessitent beaucoup plus de paramètres, ce qui les rend moins adaptés au traitement d'images complexes. Ainsi, les CNN sont particulièrement efficaces pour des applications comme la reconnaissance d'images et la détection d'objets, là où les relations spatiales et les motifs locaux jouent un rôle crucial.

3.2.4.2 Algorithme utilisé

1 Détection de Visage : Pour commencer, nous devons localiser les visages dans une image. Nous utilisons la fonction `facerecognition.facelocations`, qui repose sur l'algorithme Histogram of Oriented Gradients (HOG). Cet algorithme analyse les gradients d'orientation pour détecter les formes de visage. C'est une méthode efficace qui fonctionne bien même avec des variations de lumière et de pose.

2 Extraction des Caractéristiques : Une fois les visages détectés, nous passons à l'extraction des caractéristiques. La fonction `facerecognition.faceencodings` génère des vecteurs de caractéristiques pour chaque visage. Ces vecteurs sont créés à l'aide de réseaux de neurones convolutifs (CNN), qui sont très performants pour capturer les détails uniques de chaque visage. Ces vecteurs, ou embeddings, représentent les traits distinctifs du visage de manière numérique.

3 Comparaison des Visages : La dernière étape consiste à comparer les vecteurs de caractéristiques des visages pour déterminer leur similitude. Nous utilisons la distance euclidienne, calculée avec `np.linalg.norm`, pour mesurer la proximité entre deux vecteurs. Si la distance est inférieure à un seuil défini (0.6 dans notre cas), les visages sont considérés comme une correspondance.

3.2.5 Réalisation du Sprint 1

Voici des exemples d'interfaces graphiques créées au cours de ce sprint, illustrés par les figures ci-dessous

- **Interface login pour l'application Mobile**

Avant d'accéder à l'application, l'Agent de terrain doit s'authentifier en saisissant son email et son mot de passe dans le formulaire de connexion. Une erreur est générée si les champs obligatoires sont vides ou si l'utilisateur entre des informations incorrectes.

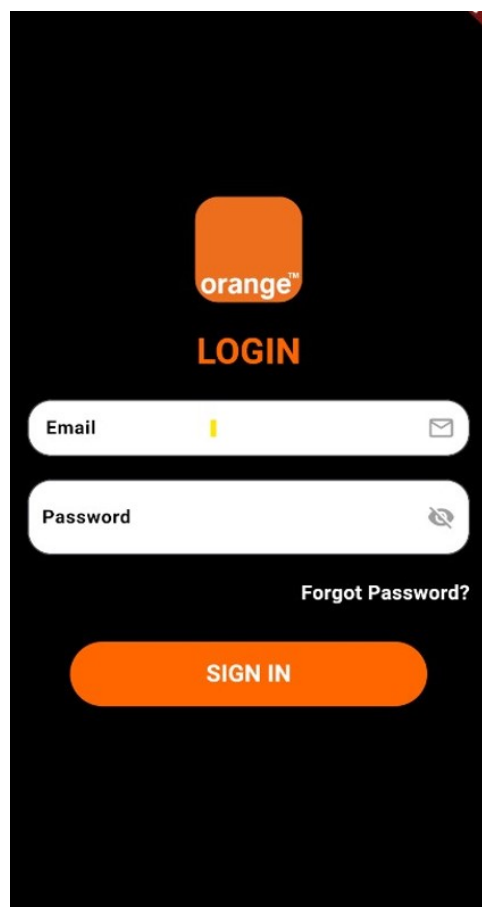


Figure 3.9 – Interface login

- **Interface Pointage pour l'application Mobile**

Lors de l'authentification, l'agent de terrain accède à une interface de pointage qui nécessite l'activation de sa localisation et la soumission de son image pour effectuer le pointage.

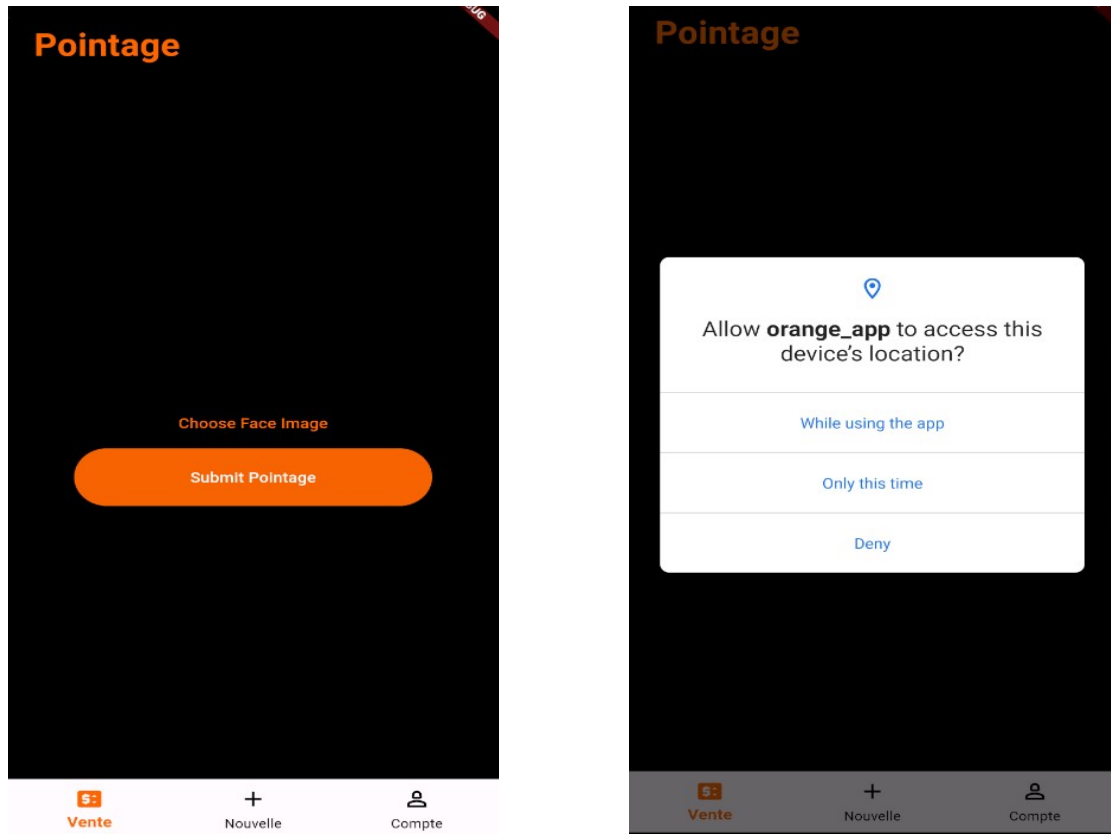


Figure 3.10 – Interface Pointage d'application mobile

- **Interface AjouterVente pour l'application Mobile**

Avant d'ajouter une vente, l'agent de terrain doit remplir le formulaire de vente et y joindre les images.

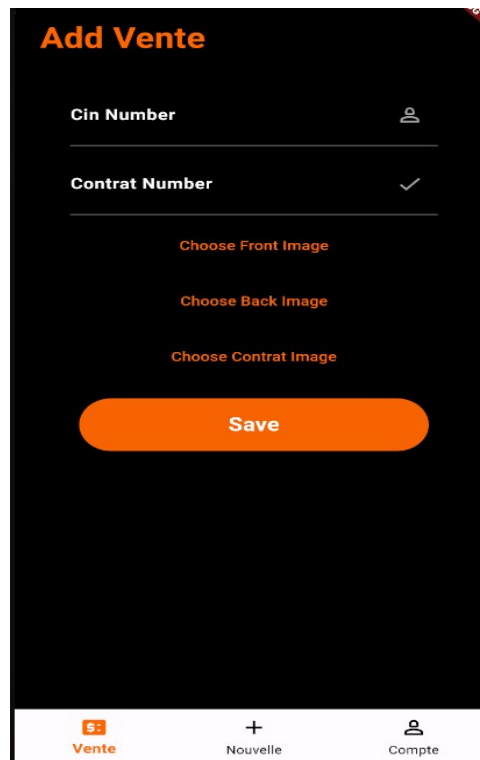


Figure 3.11 – Interface AjouterVente pour l’application Mobile

○ **Interface de la liste des ventes pour l’application Mobile**



Figure 3.12 – Interface de la liste des ventes pour l’application Mobile

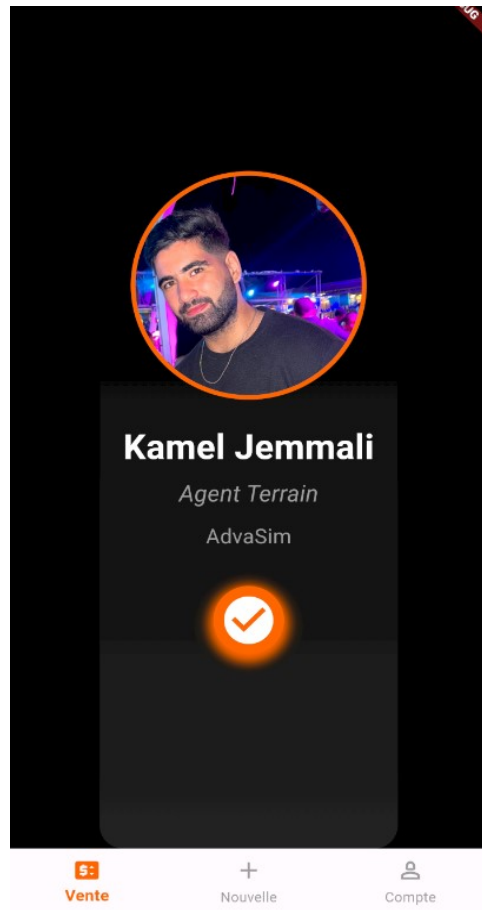


Figure 3.13 – Interface du compte d'utilisateur

3.3 Conclusion

Ce chapitre s'est concentré sur le développement et l'intégration de l'application mobile destinée aux agents de terrain. Nous avons décrit les cas d'utilisation, les classes et les séquences qui facilitent le travail des agents sur le terrain. En particulier, nous avons abordé l'intégration de la reconnaissance faciale et de SignalR. La reconnaissance faciale assure une vérification rapide et sécurisée des identités lors du pointage, tandis que SignalR permet une communication en temps réel entre les agents et le système central. Ces technologies innovantes renforcent la sécurité et l'efficacité du pointage, tout en améliorant la réactivité des agents de terrain.

Chapitre 4 : Etude et réalisation du sprint2

4.1 Introduction

Ce chapitre va présenter le deuxième sprint « Agent Saisie », en détaillant les tâches essentielles associées à chaque User Story. Ces tâches seront accompagnées de diagrammes UML qui illustreront la conception et la logique appliquées au cours des différentes itérations.

4.2 Sprint 2 : Agent Saisie

Les agents de saisie sont responsables de l'entrée et de la validation des données dans le système. Ce chapitre se concentre sur les outils et les fonctionnalités disponibles pour ces agents afin de faciliter leur travail quotidien. Nous aborderons la manière dont les agents de saisie peuvent valider ou refuser les ventes, consulter la liste des pointages, et s'assurer que toutes les informations sont correctement enregistrées et mises à jour. Le rôle des agents de saisie est crucial pour maintenir l'intégrité et l'exactitude des données au sein du système.

4.2.1 Backlog du Sprint 2

Le tableau ci-dessous donne un aperçu détaillé sur le Backlog du deuxième sprint qui prend en charge plusieurs fonctionnalités d'agent saisie.

ID	Histoire	Estimation/jours
1	En tant qu'agent saisie, je peux s'authentifier	2
2	En tant qu'agent saisie,j'ai la possibilité de Contrôler les pointages	1
3	En tant qu'agent saisie ,j'ai la possibilité de contrôler les ventes	3
4	En tant qu'agent saisie ,j'ai la possibilité de valider une vente	5
5	En tant qu'agent saisie j'ai la possibilité de refuser une vente	2
6	En tant qu'agent saisie ,j'ai la possibilité de télécharger le tableau de vente (Excel)	1
7	En tant qu'agent saisie ,j'ai la possibilité de télécharger le tableau de pointage (PDF - Excel)	1

Table 4.1 – Tableau des histoires et estimations en jours du deuxième sprint

4.2.2 Spécification fonctionnelle du Sprint2

4.2.2.1 Diagramme de cas d'utilisation du Sprint 2

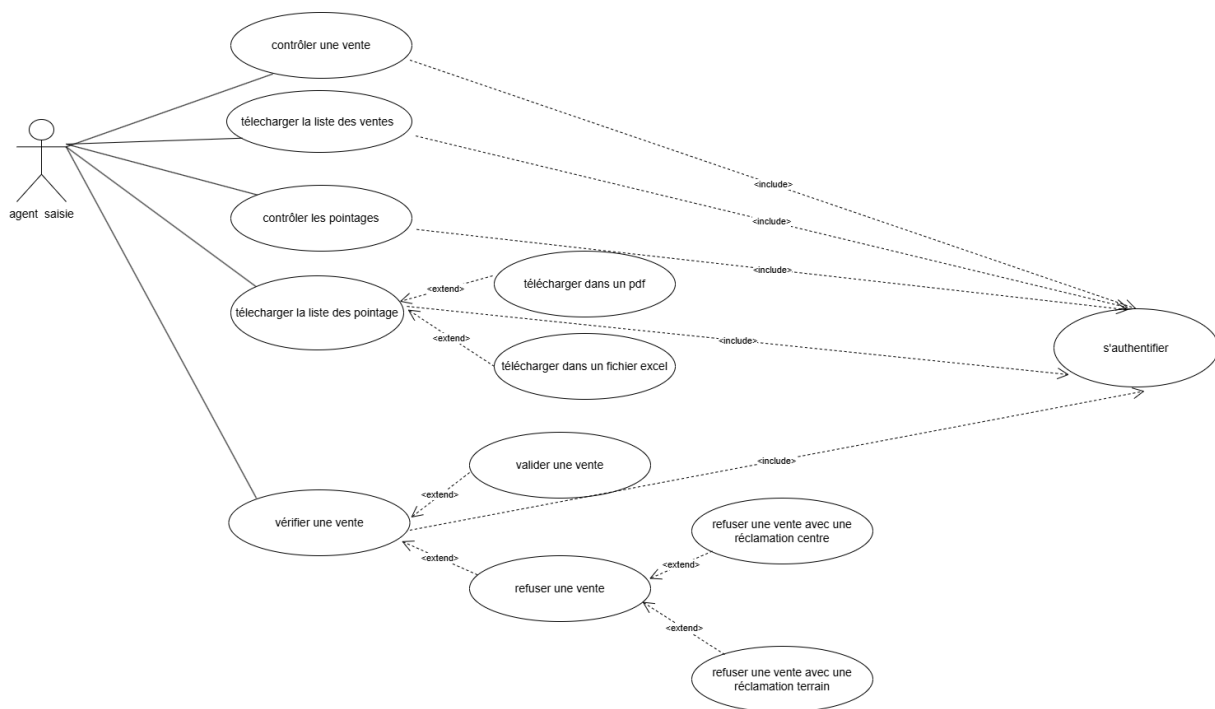


Figure 4.1 – Diagramme de cas d'utilisation pour « Agent Saisie »

4.2.2.2 Description textuelle de cas d'utilisation

«Refuser une vente»

Titre	Refuser une vente
Acteur	Agent saisie
Précondition	Agent de Saisie authentifié avec succès
Scénario Nominal	<ol style="list-style-type: none">1. L'Agent de Saisie consulte le menu des ventes.2. L'Agent de Saisie sélectionne une vente à traiter.3. L'Agent de Saisie examine toutes les données du client ainsi que les images.4. Lorsqu'un Agent de Saisie refuse une vente, il doit cliquer sur "Refuser", décrire le problème dans une fenêtre pop-up et confirmer via "Réclamation Terrain" ou "Réclamation Centre".5. Le système renvoie ensuite à la liste mise à jour des ventes
Post Condition	La vente refusée s'affiche en rouge dans le tableau.

Table 4.2 – Refuser une vente

«Valider une vente»

Titre	Valider une vente
Acteur	Agent de Saisie
Précondition	Agent de Saisie authentifié avec succès
Scénario Nominal	<ol style="list-style-type: none">1. L'Agent de Saisie consulte le menu des ventes2. L'Agent de Saisie sélectionne une vente à traiter3. L'Agent de Saisie examine toutes les données du client ainsi que les images.4. Quand l'Agent de Saisie valide une vente, les images du recto et du verso de la carte d'identité (frontcin et backcin) sont fusionnées puis sauvegardées dans un dossier nommé selon l'ID de la vente.5. Le système renvoie ensuite à la liste des ventes
Post Condition	La vente validée s'affiche en vert dans le tableau.

Table 4.3 – Valider une vente

«Contrôler le pointage»

Titre	Contrôler le pointage
Acteur	Agent de Saisie
Précondition	Agent de Saisie authentifié avec succès
Scénario Nominal	<ol style="list-style-type: none">1. L'Agent de Saisie consulte le menu des pointages2. L'Agent de Saisie sélectionne un pointage3. L'Agent de Saisie examine l'image, la date et la localisation d'agent terrain.
Post Condition	L'Agent de Saisie accède le menu des pointages une autre fois.

Table 4.4 – Contrôler le pointage

- **Utilité de l'Enregistrement des Données de Pointage et de Vente dans des Formats PDF ou Excel**

L'enregistrement des données de pointage et de vente dans des formats tels que PDF ou Excel offre plusieurs avantages stratégiques et opérationnels qui sont essentiels pour la gestion efficace et la documentation dans notre organisation.

- **Accessibilité et Partageabilité** : Les formats PDF et Excel sont largement utilisés et reconnus dans le monde professionnel. Enregistrer les informations dans ces formats permet une meilleure accessibilité et facilite le partage des données entre les différentes parties prenantes, y compris la gestion interne, les auditeurs, et les régulateurs.
- **Intégrité et Sécurité des Données** : Le format PDF est particulièrement utile pour préserver l'intégrité des données. Une fois les données converties en PDF, elles ne peuvent pas être facilement modifiées sans laisser de traces, ce qui est crucial pour le maintien des archives officielles et la conformité réglementaire. Excel, quant à lui, permet une manipulation et une analyse plus dynamiques des données, tout en offrant des options pour sécuriser les données, comme le verrouillage des cellules ou la protection par mot de passe.
- **Analyse et Reporting** : Excel est exceptionnellement puissant pour l'analyse de données grâce à ses fonctions avancées de tri, de filtrage, et de calcul. Utiliser Excel pour enregistrer les ventes et les pointages permet d'effectuer des analyses périodiques et de générer des rapports de performance, des tendances de vente, et d'autres insights opérationnels qui sont essentiels pour la prise de décision stratégique.
- **Archivage et Référence** : Le stockage des données en formats PDF et Excel facilite l'archivage et la création d'une base de données historique. Cela permet aux entreprises de conserver un historique consultable des activités de vente et de pointage, essentiel pour les audits internes ou externes, les revues de performance, et la planification stratégique.

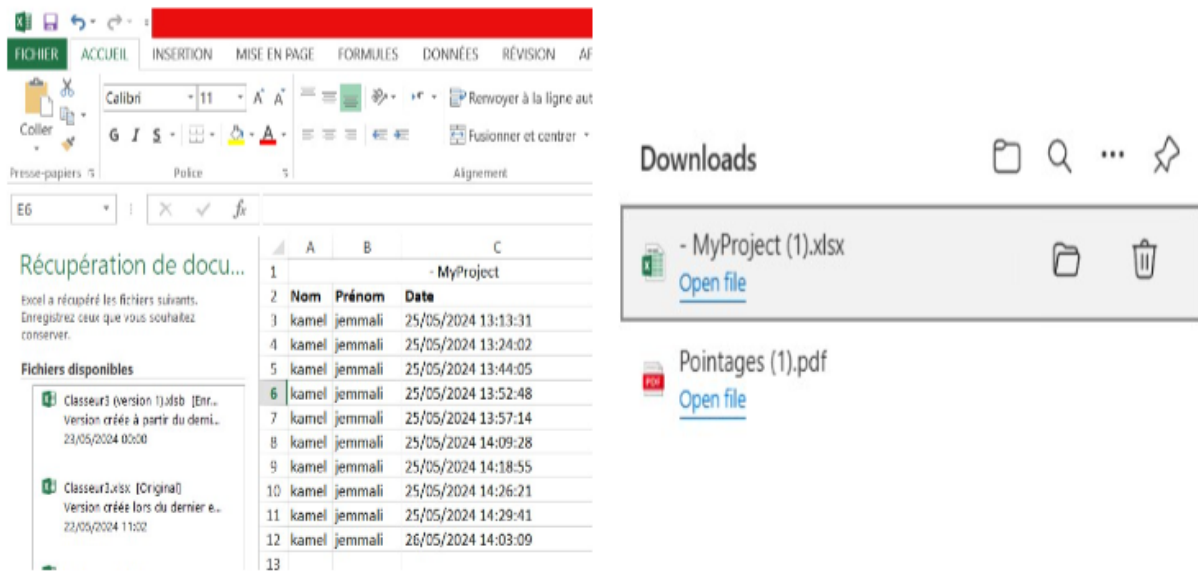


Figure 4.2 – télécharger la liste des ventes et pointage (excel-pdf)

4.2.2.3 Diagramme de classe du Sprint 2

Le diagramme de classes est un outil permettant de représenter la structure interne d'un système en exposant les différentes classes, leurs attributs, ainsi que les relations structurelles qui les lient.

La figure ci-dessous illustre le diagramme de classes que nous avons utilisé pour développer le deuxième sprint

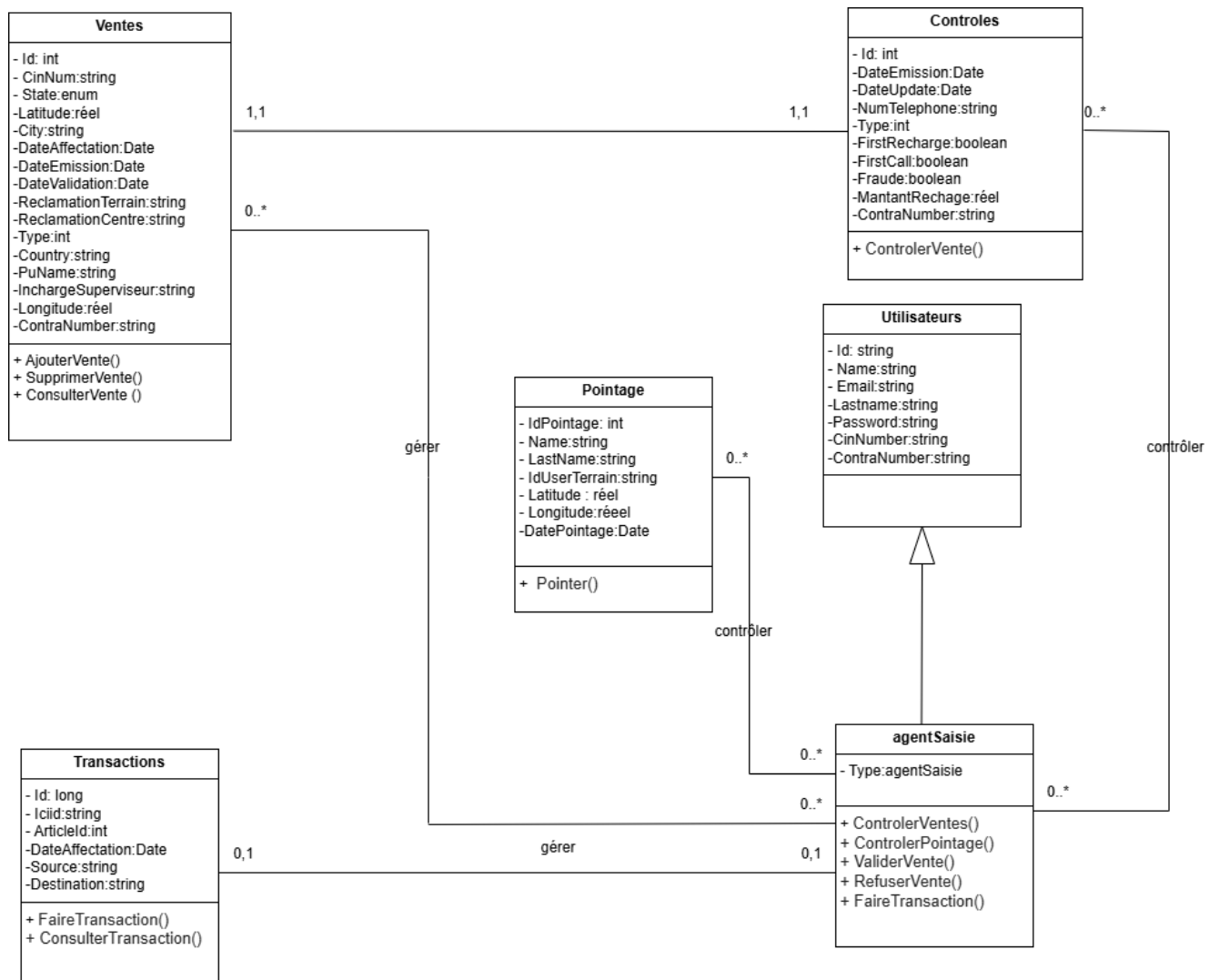


Figure 4.3 – diagramme de classe du sprint 2

4.2.2.4 Diagramme de séquence

Afin de représenter les communications internes de notre système informatique et avec les acteurs externes, nous employons le diagramme de séquence UML. Ce diagramme permet de visualiser de manière chronologique les interactions entre les objets et les acteurs, ainsi que les messages échangés entre eux.

○ Diagramme de séquence détaillée(Valider ou Refuser une vente)

Ce diagramme de séquence UML illustre le processus de validation ou de refus d'une vente par un Agent de Saisie, soulignant les interactions entre divers composants du système : L'Agent de Saisie examine les données d'une vente sur l'interface, incluant les images de la carte d'identité

(CIN), le contrat... Si un problème est détecté, l'Agent refuse la vente et un pop-up s'ouvre, permettant à l'Agent de saisir une réclamation, soit de type Terrain, soit de type Centre. En cas de refus, la vente apparaît dans le tableau sous la réclamation choisie, avec une description expliquant pourquoi le refus a eu lieu. Le statut passe de 1 (vente en cours de traitement) à 3 (réclamation Centre) ou 4 (réclamation Terrain), les modifications sont enregistrées dans la base de données, et l'Agent est redirigé vers la page des ventes. Si les données sont valides, le statut devient 2 (validé), la modification est sauvegardée dans la base de données, et l'Agent est également redirigé vers la page des ventes.

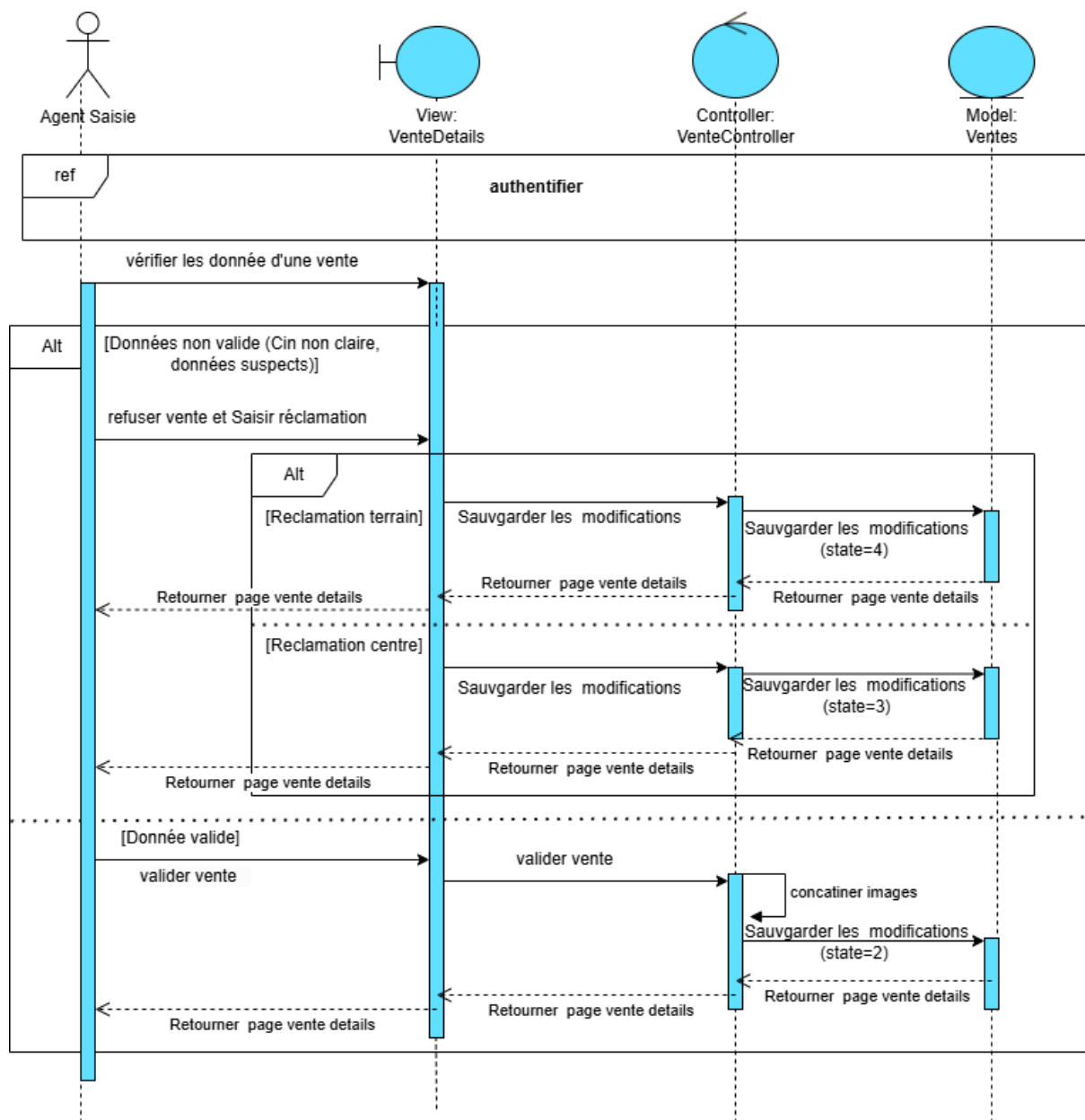


Figure 4.4 – Diagramme de sequence valider/refuser vente

4.2.3 Réalisation du Sprint 2

Les figures ci-dessous illustrent des exemples d'interfaces graphiques développées au cours de ce sprint.

The top screenshot shows a search form with the following fields: **icc-id**, **cin**, **de** (mm/dd/yyyy), and **a** (mm/dd/yyyy). Below the search fields are checkboxes for **non traité**, **en cours de traitement**, **validée**, **standard**, **problème centrale**, and **problème terrains**. There are buttons for **Excel** and **Rechercher**.

The table below shows the results of the search:

Numéro d'ordre	Cin / Identifiant	Date d'émission	Type	Etat	ReclamationCenter	ReclamationTerrain	Icc-Id	Article	
1	14141414	21/05/2024 00:00:00	standard	Refusé Center	reclamation centre images floues		510	Unknown Article	Details delete
2	14141414	21/05/2024 00:00:00	standard	traité			510	Unknown Article	Details delete
3	14141414	21/05/2024 00:00:00	standard	en cours de traitement			510	Unknown Article	Details delete
4	14141414	25/05/2024 00:00:00	standard	Refusé Terrain		reclamation terrain	510	Unknown Article	Details delete

The bottom screenshot shows a more detailed search form with the following fields: **icc-id**, **cin**, **de** (mm/dd/yyyy), **a** (mm/dd/yyyy), and **User Type** (Agent). Below the search fields are checkboxes for **non traité**, **en cours de traitement**, **validée**, **soumis Picasso**, **problème centrale**, **problème terrains**, **prtb non traitée**, **prtb en cours de traitement**, **prtb validée**, **prtb soumis picasso**, **Standard**, **Portabilité**, **Etranger**, **Aéroport standard**, and **Aéroport étranger**. There are buttons for **Excel** and **Rechercher**.

The table below shows the results of the search:

Numéro d'ordre	Cin / Identifiant	Date d'émission	Type	Etat	ReclamationCenter	ReclamationTerrain	Icc-Id	PR	Gouvernement	IdUserTerrain
----------------	-------------------	-----------------	------	------	-------------------	--------------------	--------	----	--------------	---------------

Figure 4.5 – Interfaces pour la liste des ventes

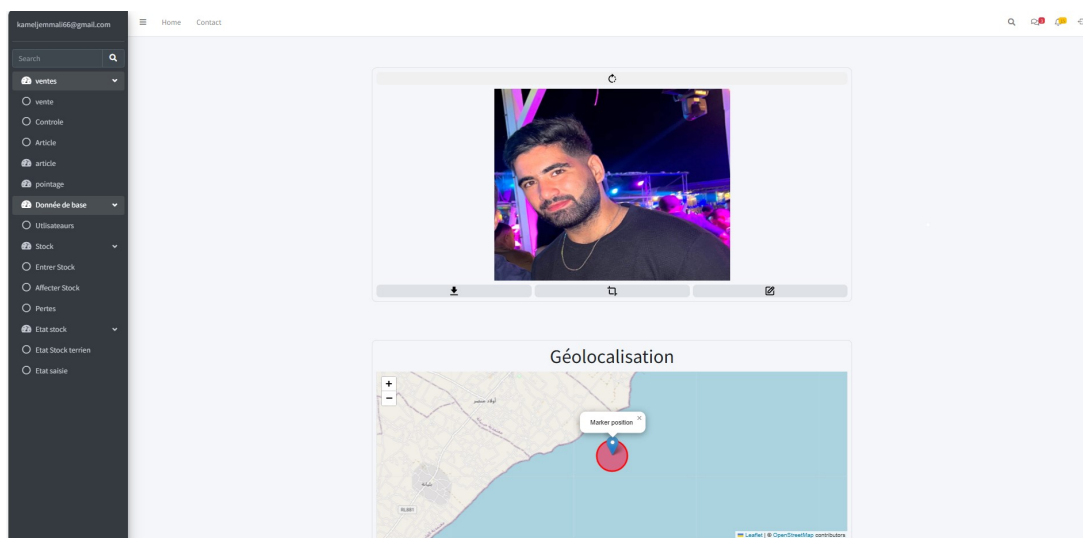
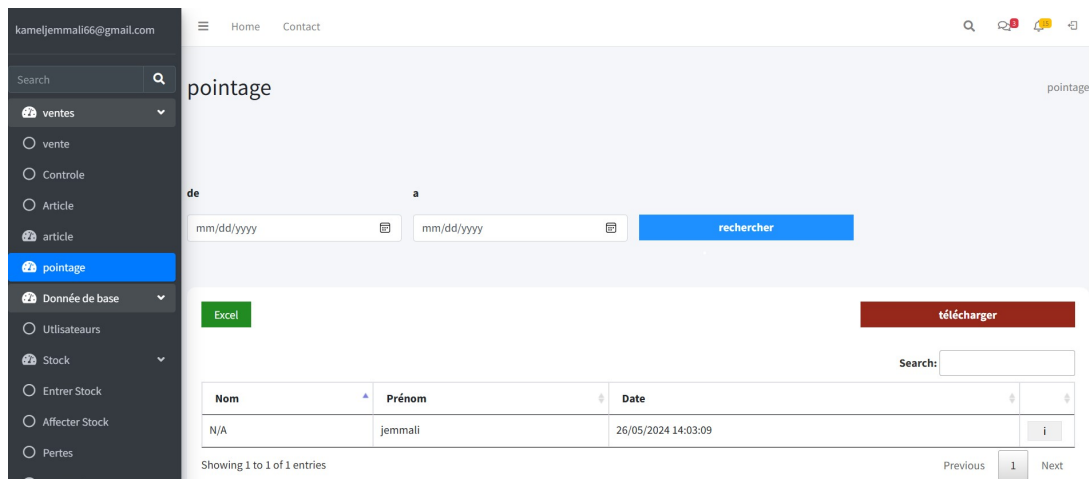


Figure 4.6 – Interfaces pour la liste des pointages du jour et pointage details

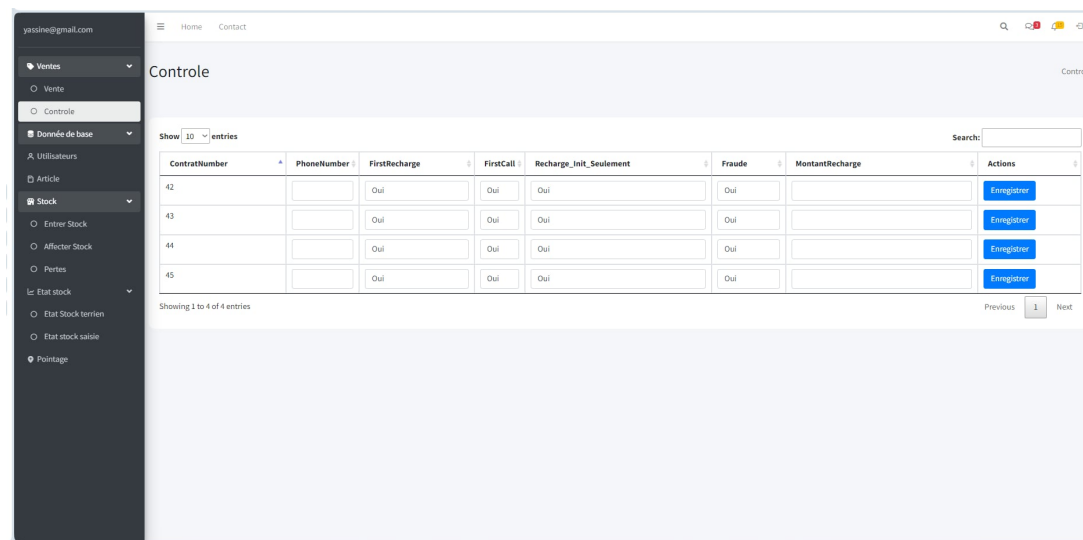


Figure 4.7 – Interfaces controle vente

Pour effectuer un contrôle efficace des puces vendues, nous suivons une procédure détaillée et systématique. Nous commençons par rassembler toutes les puces vendues et les identifions avec leurs numéros de contrat et de téléphone respectifs. Ensuite, nous vérifions si la première recharge a été effectuée pour chaque puce, ce qui est indiqué dans la colonne "FirstRecharge". Une valeur "Oui" confirme que la recharge initiale a été réalisée. Nous validons également si un premier appel a été passé à partir de chaque puce en vérifiant la colonne "FirstCall". La colonne "RechargeInitSeulement" doit indiquer "Oui" pour s'assurer que seules les recharges initiales ont été effectuées. Nous examinons chaque puce pour toute activité suspecte ou frauduleuse en inspectant la colonne "Fraude". Un "Oui" ici signifie qu'une fraude a été détectée et nécessite une enquête approfondie. Le montant de la recharge est inscrit dans la colonne "MontantRecharge" pour chaque puce, afin de vérifier les transactions et les crédits disponibles. Après avoir rempli toutes les informations nécessaires, nous cliquons sur le bouton "Enregistrer" pour chaque entrée afin de sauvegarder les données et valider les transactions. Enfin, nous documentons toutes les étapes et les résultats du contrôle, créant un rapport détaillé pour chaque puce, incluant les recharges, les appels, et les éventuelles fraudes détectées. En suivant ces étapes, nous pouvons assurer une gestion rigoureuse et précise des puces vendues, en vérifiant que toutes les transactions sont valides et en détectant toute activité suspecte. Cela garantit la satisfaction des

clients et maintient l'intégrité de notre système de distribution.

4.3 Conclusion

Ce chapitre a été consacré à l'analyse du rôle de l'agent de saisie dans notre système. À travers des diagrammes de cas d'utilisation, des diagrammes de classes et des diagrammes de séquence, nous avons illustré comment l'agent de saisie interagit avec le système pour enregistrer et valider les ventes. Cette analyse nous a permis de définir clairement les processus et les flux de travail, assurant ainsi l'efficacité et l'exactitude des données saisies. L'agent de saisie est essentiel pour la collecte de données précises, ce qui est fondamental pour la prise de décisions informées et la gestion efficace des ventes.

Chapitre 5 : Etude et réalisation du sprint3

5.1 Introduction

Ce chapitre va présenter le troisième sprint « Agent Stock », en détaillant les tâches essentielles associées à chaque User Story. Ces tâches seront accompagnées de diagrammes UML qui illustreront la conception et la logique appliquées au cours des différentes itérations.

5.2 Sprint 3 : « Agent Stock »

Les agents de stock sont chargés de la gestion des inventaires et de la surveillance des articles. Ce chapitre explore les fonctionnalités mises à disposition des agents de stock pour gérer les entrées et sorties de marchandises, suivre les niveaux de stock en temps réel, et déclarer les pertes ou les anomalies. Nous discuterons également des méthodes utilisées pour assurer une gestion précise et efficace des stocks, ce qui est essentiel pour le bon fonctionnement des opérations commerciales.

5.2.1 Backlog du Sprint 3

Le tableau ci-dessous donne un aperçu détaillé sur le Backlog du troisième sprint qui prend en charge plusieurs fonctionnalités d'agent stock. «Agent Stock»

ID	Histoire	Estimation/jours
1	En tant qu'agent stock , je peux s'authentifier	2
2	En tant qu'agent stock , j'ai la possibilité d'ajouter un article	1
3	En tant qu'agent stock , j'ai la possibilité de modifier un article	1
4	En tant qu'agent stock , j'ai la possibilité de supprimer un article	1
5	En tant qu'agent stock , j'ai la possibilité de consulter tous les articles	1
6	En tant qu'agent stock , j'ai la possibilité d'entrer le stock	4
7	En tant qu'agent stock , j'ai la possibilité d'affecter le stock	6
8	En tant qu'agent stock,j'ai la possibilité de consulter l'état du stock	3

Table 5.1 – Tableau des histoires et estimations en jours du sprint 3

5.2.2 Spécification fonctionnelle du Sprint 3

Cette section introduit la phase d'analyse, centrée sur la fonctionnalité de l'agent de Stock en répondant à la question : "Que fait l'agent de Stock ?" Les réponses sont illustrées à travers un diagramme de cas d'utilisation, complété par des descriptions textuelles détaillées pour chaque cas d'utilisation.

5.2.2.1 Diagramme de cas d'utilisation du Sprint 3



Figure 5.1 – Diagramme de cas d'utilisation d'un agent stock

5.2.2.2 Description textuelle de cas d'utilisation

«Ajouter un article»

Titre	Ajouter un article
Acteur	Agent Stock
Précondition	Agent de stock authentifié avec succès
Scénario Nominal	1.L'Agent de stock consulte le menu des articles. 2.-L'Agent de stock clique sur ajouter article 3.L'Agent de stock entre le nom d'article et sélectionne une image (l'image est obligatoire). 4.- Le système renvoie ensuite à la liste mise à jour des articles.
Post Condition	L'article est ajouté dans le tableau.

Table 5.2 – Ajouter un article

«Affecter le stock»

Titre	Affecter le stock
Acteur	Agent de stock
Précondition	Agent de stock authentifié avec succès
Scénario Nominal	<ol style="list-style-type: none">1. L'Agent de stock consulte le formulaire d'affectation.2. L'Agent de stock choisit un article, son Id et la quantité pour l'affecter à un agent de terrain ou un superviseur.3. L'Agent de stock clique sur le bouton vert pour vérifier les informations.4. Si l'article existe, le système renvoie sa source (la personne qui possédait cet article).5. L'agent de stock entre la destination (la personne à qui il veut donner cet article).6. système renvoie un message de confirmation.7. L'article est destiné à une autre personne.
Post Condition	Le système renvoie l'interface affecter stock une autre fois.

Table 5.3 – Affecter un stock

«Gérer les Transactions»

Titre	Gérer les Transactions
Acteur	Agent de Stock
Précondition	Agent de Stock authentifié et accès au système de gestion des stocks
Scénario Nominal	<ol style="list-style-type: none">1. L'Agent de stock consulte le formulaire d'entrée de stock.2. L'Agent de stock sélectionne un article et saisit les données.3. L'Agent de stock clique sur le bouton vert pour vérifier les informations.4. Si les informations sont correctes, le système affiche un message de confirmation.5. L'Agent de stock clique sur le bouton de confirmation pour finaliser l'entrée de stock.6. Le système enregistre l'entrée de stock et affiche une confirmation de réussite.
Post Condition	Le système retourne à l'interface d'entrée de stock prête pour une nouvelle entrée.

Table 5.4 – Gérer les Transactions

5.2.2.3 Diagramme de classe du Sprint 3

Le diagramme de classes est un outil permettant de représenter la structure interne d'un système en exposant les différentes classes, leurs attributs, ainsi que les relations structurelles qui les lient.

La figure ci-dessous illustre le diagramme de classes que nous avons utilisé pour développer le troisième sprint.

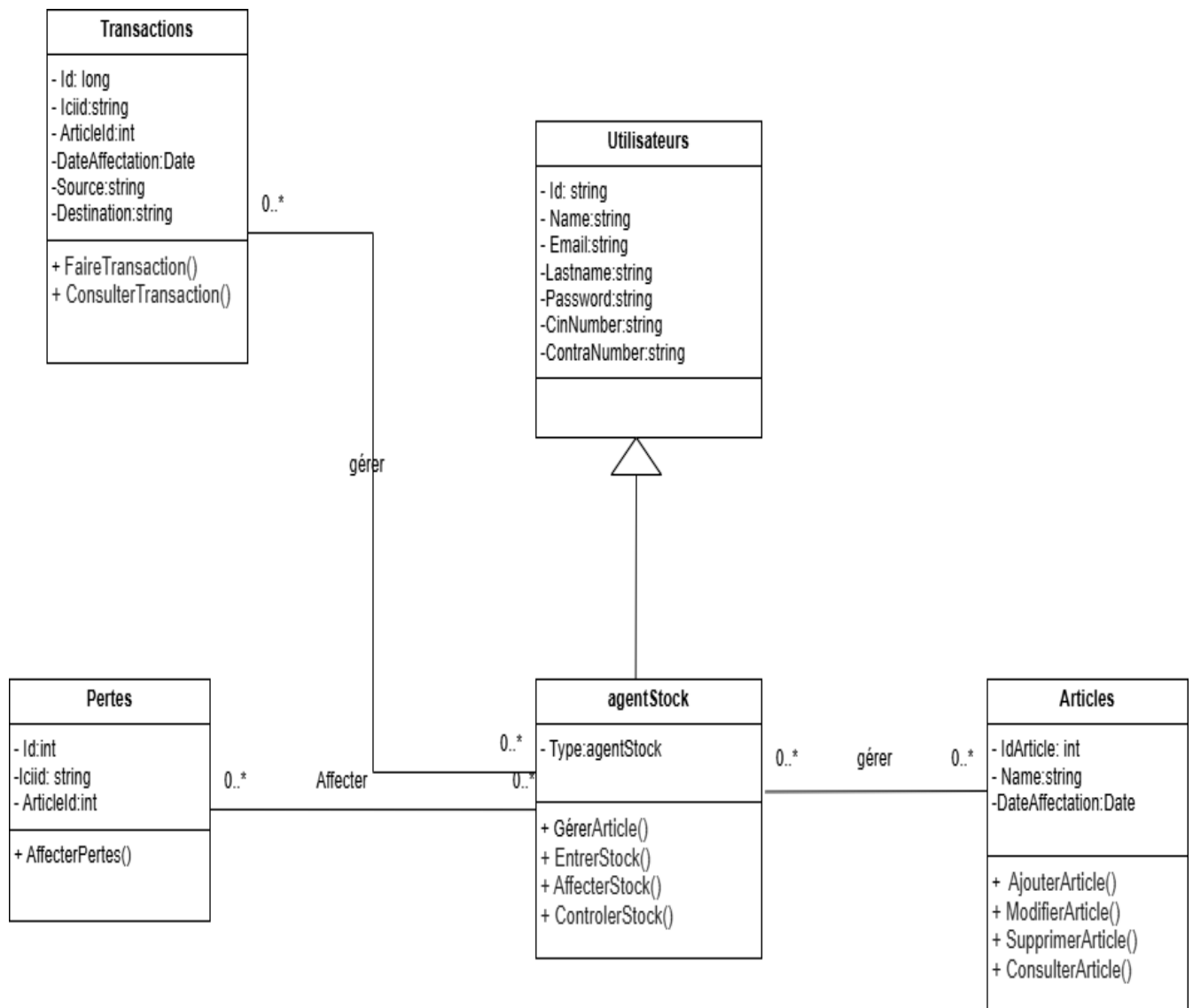


Figure 5.2 – Diagramme de classe « Agent Stock »

5.2.2.4 Diagramme de séquence

Afin de représenter les communications internes de notre système informatique et avec les acteurs externes, nous employons le diagramme de séquence UML. Ce diagramme permet de visualiser de manière chronologique les interactions entre les objets et les acteurs, ainsi que les messages échangés entre eux.

- **Diagramme de séquence détaillée(Affecter Stock)**

L'agent de stock commence par une authentification. Une fois connecté, il peut affecter le stock d'un article en saisissant des informations telles que l'ICCID (chaque article possède deux ICCID), le nom de l'article (le système présente une liste de tous les articles disponibles) et la quantité. Si l'article est disponible et que la quantité est correcte (la différence entre ICCID2 et ICCID1), le système affiche à nouveau la page d'affectation du stock. Il montre alors la source et le nom de l'article et demande à l'agent de stock d'entrer la destination. Si toutes les données sont valides, le système affiche un message de confirmation et renvoie à l'interface d'affectation du stock. En cas d'erreur, le système affiche un message d'échec et redirige vers la même interface

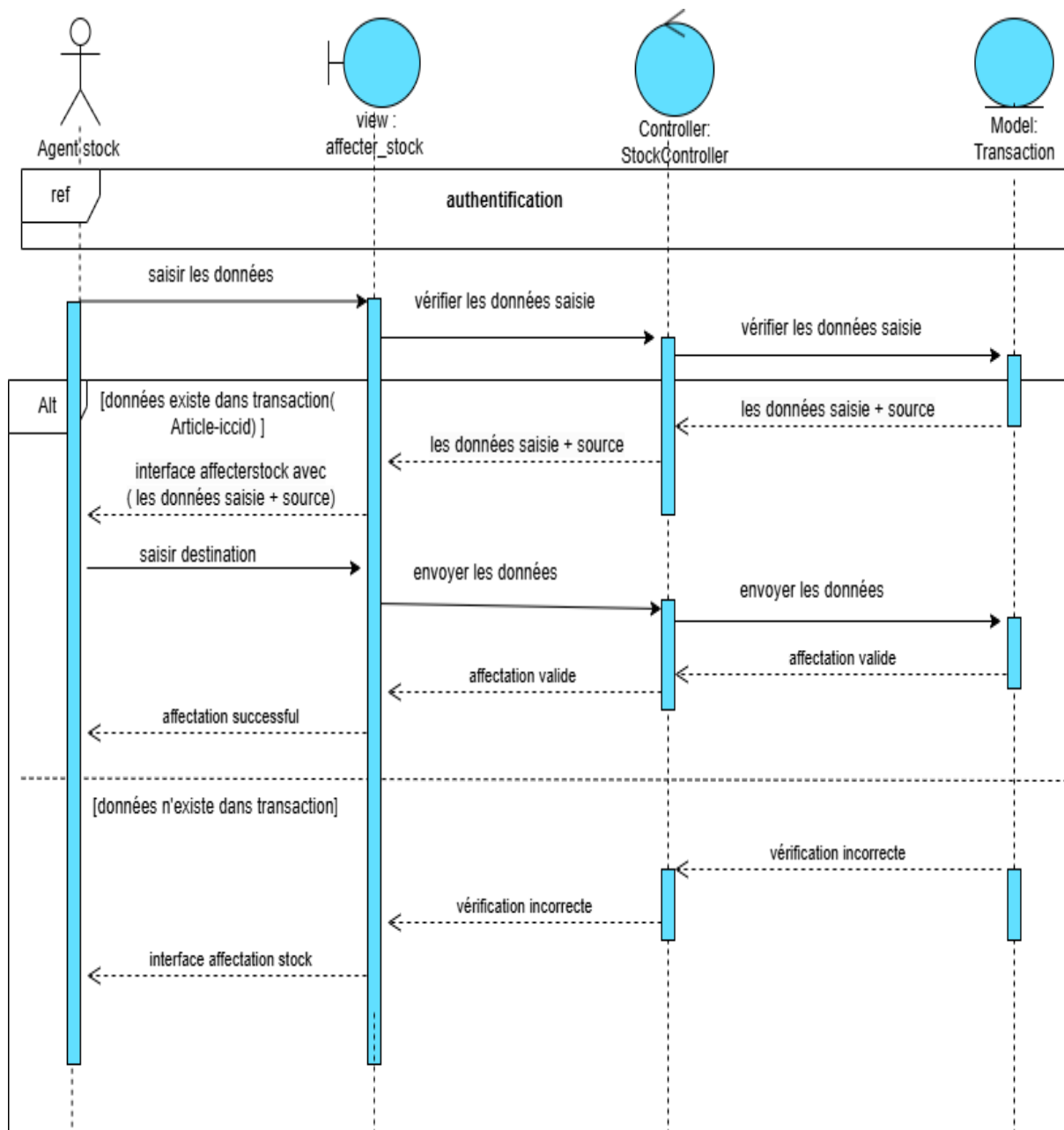


Figure 5.3 – Diagramme de séquence affecter stock

○ Diagramme de séquence détaillée(Ajouter Stock)

L'agent de stock est d'abord authentifié. Ensuite, l'agent de stock peut entrer le stock d'un article en saisissant les données telles que l'ICCID (chaque article a deux ICCID), le nom de l'article(le système présente une liste de tous les articles disponibles) et la quantité. Si l'article existe et que la quantité est correcte (ICCID2 - ICCID1), le système renvoie un message de confirmation. Sinon, si les données sont invalides, le système renvoie un message d'échec.

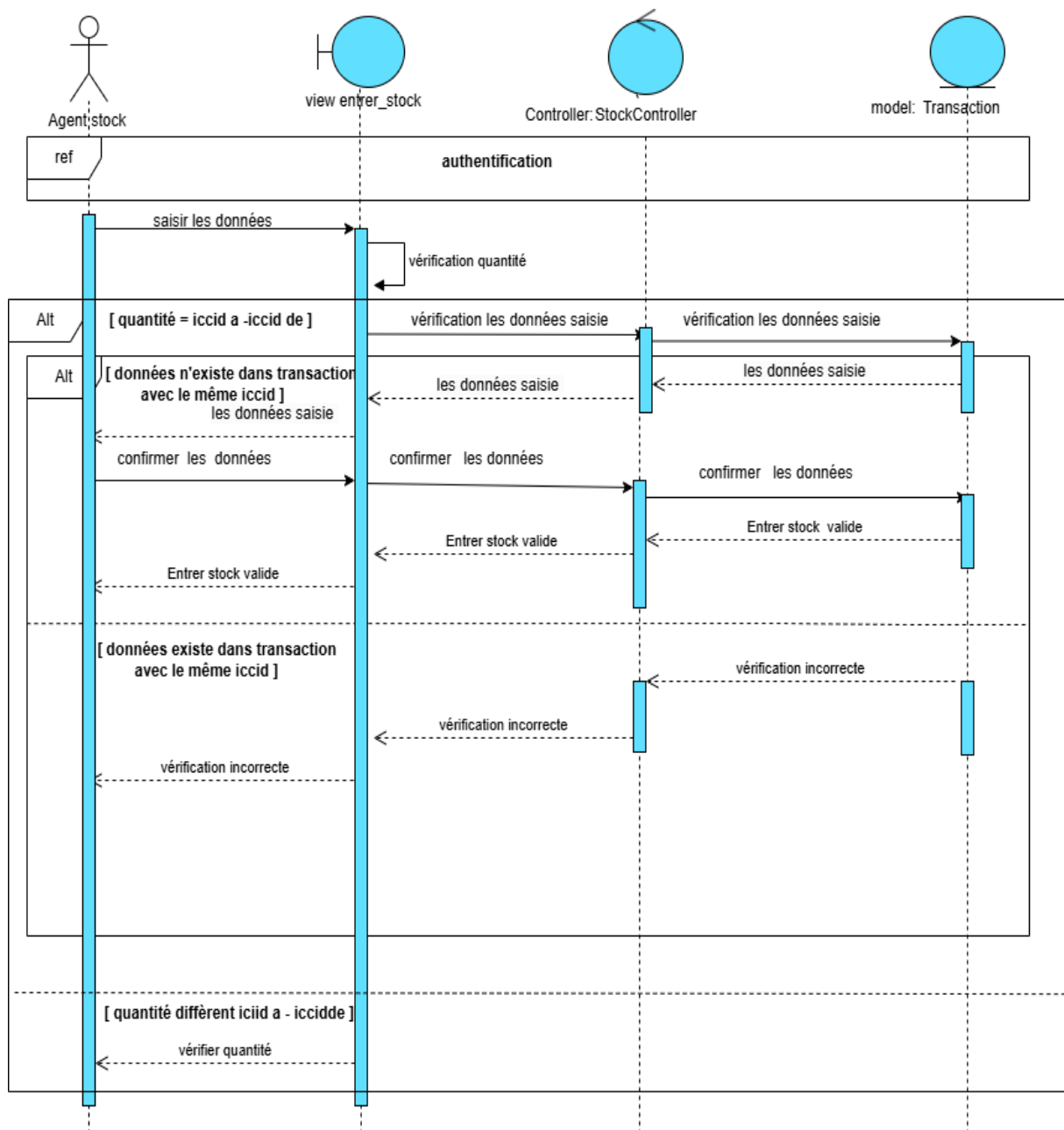


Figure 5.4 – Diagramme de séquence entrer stock

5.2.3 Réalisation du Sprint 3

Les figures ci-dessous illustrent des exemples d'interfaces graphiques développées au cours de ce sprint.

Ventes

Vente

Controle

Donnée de base

Utilisateurs

Article

Stock

Entrer Stock

Affecter Stock

Pertes

Etat stock

Etat Stock terrien

Entrée stock

confirmer

article

Select Article

Quantite Totale

lcld de

veuiller saisir le premier lcld

lcld à

veuiller saisir le dernier lcld

Figure 5.5 – Interface d’entré stock

kamel@gmail.com

Home Contact

Ventes

Vente

Controle

Donnée de base

Utilisateurs

Article

Stock

Entrer Stock

Affecter Stock

Pertes

Etat stock

Etat stock saisie

Excel

Search:

Agent	Total	Total Echouée	Total Vendu
kamel je.	1	1	0
kamel jemali	1	0	0
kamel	6	1	0

Showing 1 to 3 of 3 entries

Previous

1

Next

Figure 5.6 – Interface de l’etat stock saisie

Etat stock terrain

Stock> Etat stock terrain

Article

Excel

Search:

Superviseur	Agent	Article	Total In Stock	Total Echouée	Total Pertes	Total Vendu	Total
	magazin	2	100	0	0	0	100
	magazin	4	0	0	0	0	0
jemmali	kamel	4	99	0	1	0	99

Showing 1 to 3 of 3 entries

Previous
1
Next

Figure 5.7 – Interface de l’etat stock terrain

Ventes

Vente

Controle

Donnée de base

Utilisateurs

Article

Stock

Entrer Stock

Affecter Stock

Pertes

Etat stock

Etat Stock terrain

Etat stock saisie

Pointage

Affectation stock

Stock> Affectation stock

Article

5G

Quantite Totale

100

Iccld de

800

Iccld à

900

Ventes

Vente

Controle

Donnée de base

Utilisateurs

Article

Stock

Entrer Stock

Affecter Stock

Pertes

Etat stock

Etat Stock terrien

Etat stock saisie

Pointage

Affectation stock

confirm

destination

Source

Article

Quantite Totale

kamel

0

5G

100

lccid de

lccid à

800

900

✓

Figure 5.8 – Interface de l’affectation du stock

kamel@gmail.com

Ventes

Vente

Controle

Donnée de base

Utilisateurs

Article

Stock

Entrer Stock

Affecter Stock

Pertes

Etat stock

Etat Stock terrien

Etat stock saisie

Pointage

List_Article

List_Article

AjouterArticle

Show 10 entries

Search:

Name	Date D'Affectation	
5G	21/05/2024 01:47:11	Details Edit Delete
dallel	26/05/2024 15:13:34	Details Edit Delete
ERNSTfddg	01/05/2024 14:16:22	Details Edit Delete
yass	02/05/2024 13:38:34	Details Edit Delete

Showing 1 to 4 of 4 entries

Previous 1 Next

Figure 5.9 – Interface de liste des articles

5.3 Conclusion

L'agent de stock joue un rôle vital dans la gestion des inventaires, comme détaillé dans ce chapitre. Nous avons examiné les responsabilités de l'agent de stock à travers des diagrammes de cas d'utilisation, des diagrammes de classes et des diagrammes de séquence, mettant en évidence les processus de suivi et de mise à jour des niveaux de stock. Cette section a démontré l'importance de la gestion précise des stocks pour maintenir la disponibilité des produits et éviter les ruptures.

Chapitre 6 : Etude et réalisation du sprint4

6.1 Introduction

Ce chapitre va présenter le quatrième sprint « Administration », en détaillant les tâches essentielles associées à chaque User Story. Ces tâches seront accompagnées de diagrammes UML qui illustreront la conception et la logique appliquées au cours des différentes itérations.

6.2 Sprint 4 : Administrateur

Le rôle de l'administrateur est essentiel pour assurer le bon fonctionnement et la gestion efficace de notre système. Dans ce chapitre, nous détaillerons les responsabilités et les fonctionnalités spécifiques réservées à l'administrateur. Ce rôle inclut la gestion des utilisateurs, la surveillance des opérations, la configuration des paramètres système et la maintenance générale. L'administrateur joue un rôle central dans la supervision et l'optimisation des performances du système, garantissant ainsi une utilisation harmonieuse et sécurisée de la plateforme.

6.2.1 Backlog du Sprint 4

Le tableau ci-dessous donne un aperçu détaillé sur le Backlog du premier sprint qui prend en charge plusieurs fonctionnalités d'administrateur.

ID	Histoire	Estimation/jours
1	En tant qu'administrateur, je peux m'authentifier	2
2	En tant qu'administrateur, j'ai la possibilité de valider une vente	5
3	En tant qu'administrateur, j'ai la possibilité de refuser une vente	2
4	En tant qu'administrateur, j'ai la possibilité de supprimer une vente	1
5	En tant qu'administrateur, j'ai la possibilité de consulter toutes les ventes	3
6	En tant qu'administrateur, je peux ajouter un article	1
7	En tant qu'administrateur, je peux supprimer un article	1
8	En tant qu'administrateur, je peux modifier un article	1
9	En tant qu'administrateur, je peux consulter tous les articles	1
10	En tant qu'administrateur, j'ai la possibilité d'affecter le stock	6
11	En tant qu'administrateur, j'ai la possibilité d'entrer le stock	4
12	En tant qu'administrateur, j'ai la possibilité de contrôler une vente	3
13	En tant qu'administrateur, j'ai la possibilité d'ajouter une perte	1
14	En tant qu'administrateur, j'ai la possibilité de consulter l'état du stock terrain et saisie	3
15	En tant qu'administrateur, j'ai la possibilité de consulter la liste des pointages	3
16	En tant qu'administrateur, j'ai la possibilité de télécharger la liste de pointages et de ventes (pdf-excel)	2
17	En tant qu'administrateur, j'ai la possibilité de gérer des comptes (créer-modifier-supprimer-consulter)	6

Table 6.1 – Tableau des histoires et estimations en jours du sprint 4

6.2.2 Spécification fonctionnelle du Sprint

Cette section introduit la phase d'analyse, centrée sur la fonctionnalité de l'administrateur en répondant à la question : "Que fait l'administrateur?" Les réponses sont illustrées à travers un diagramme de cas d'utilisation, complété par des descriptions textuelles détaillées pour chaque cas d'utilisation.

Table 6.2 – Ajouter un compte

6.2.2.3 Diagramme de classe du Sprint 1

Le diagramme de classes est un outil permettant de représenter la structure interne d'un système en exposant les différentes classes, leurs attributs, ainsi que les relations structurelles qui les lient.

La figure ci-dessous illustre le diagramme de classes que nous avons utilisé pour développer le premier sprint.

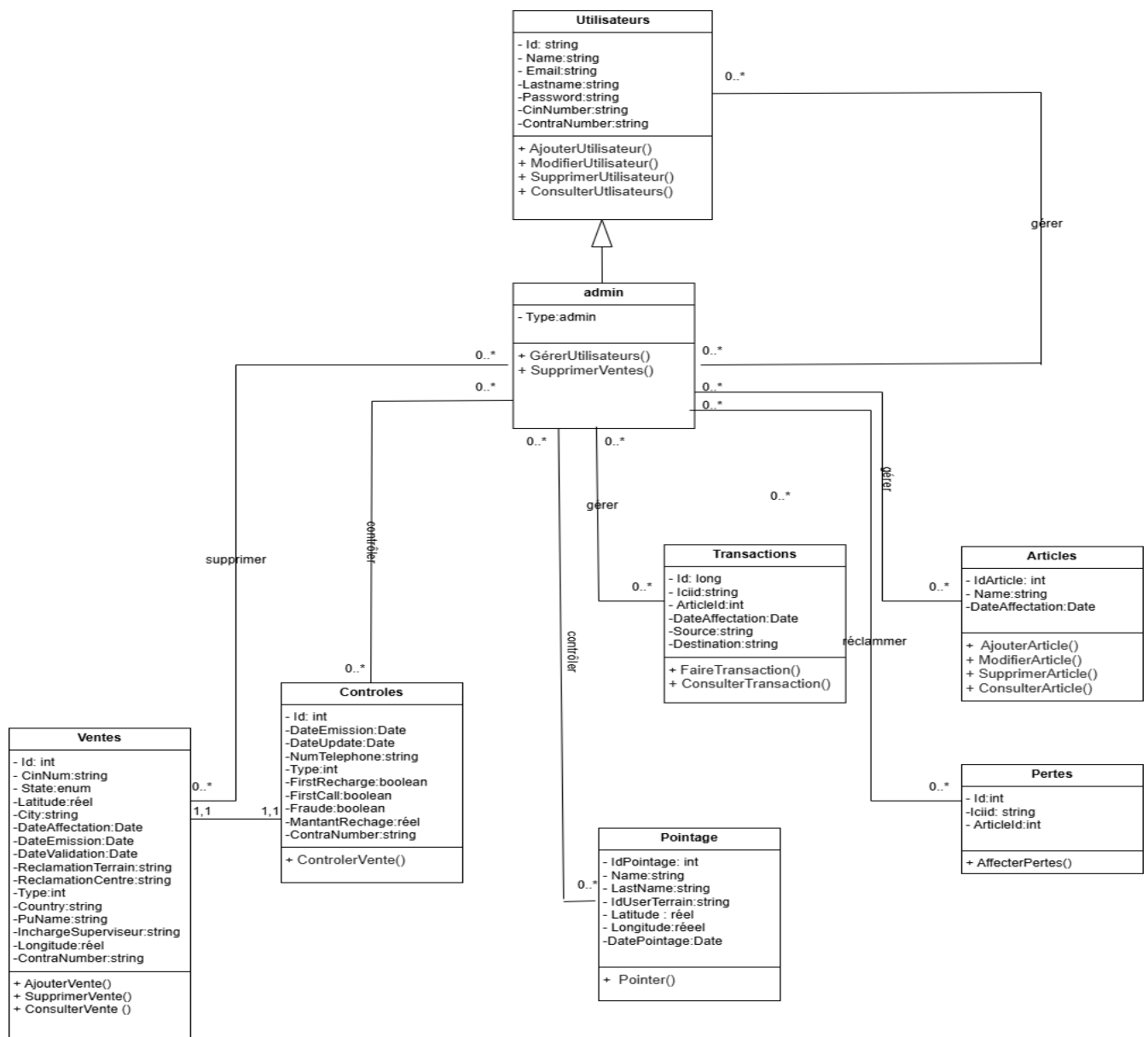


Figure 6.2 – Diagramme de classe du sprint 4

6.2.2.4 Diagramme de séquence

Afin de représenter les communications internes de notre système informatique et avec les acteurs externes, nous employons le diagramme de séquence UML. Ce diagramme permet de visualiser de manière chronologique les interactions entre les objets et les acteurs, ainsi que les messages échangés entre eux.

- **Diagramme de séquence détaillée(s'authentifier)**

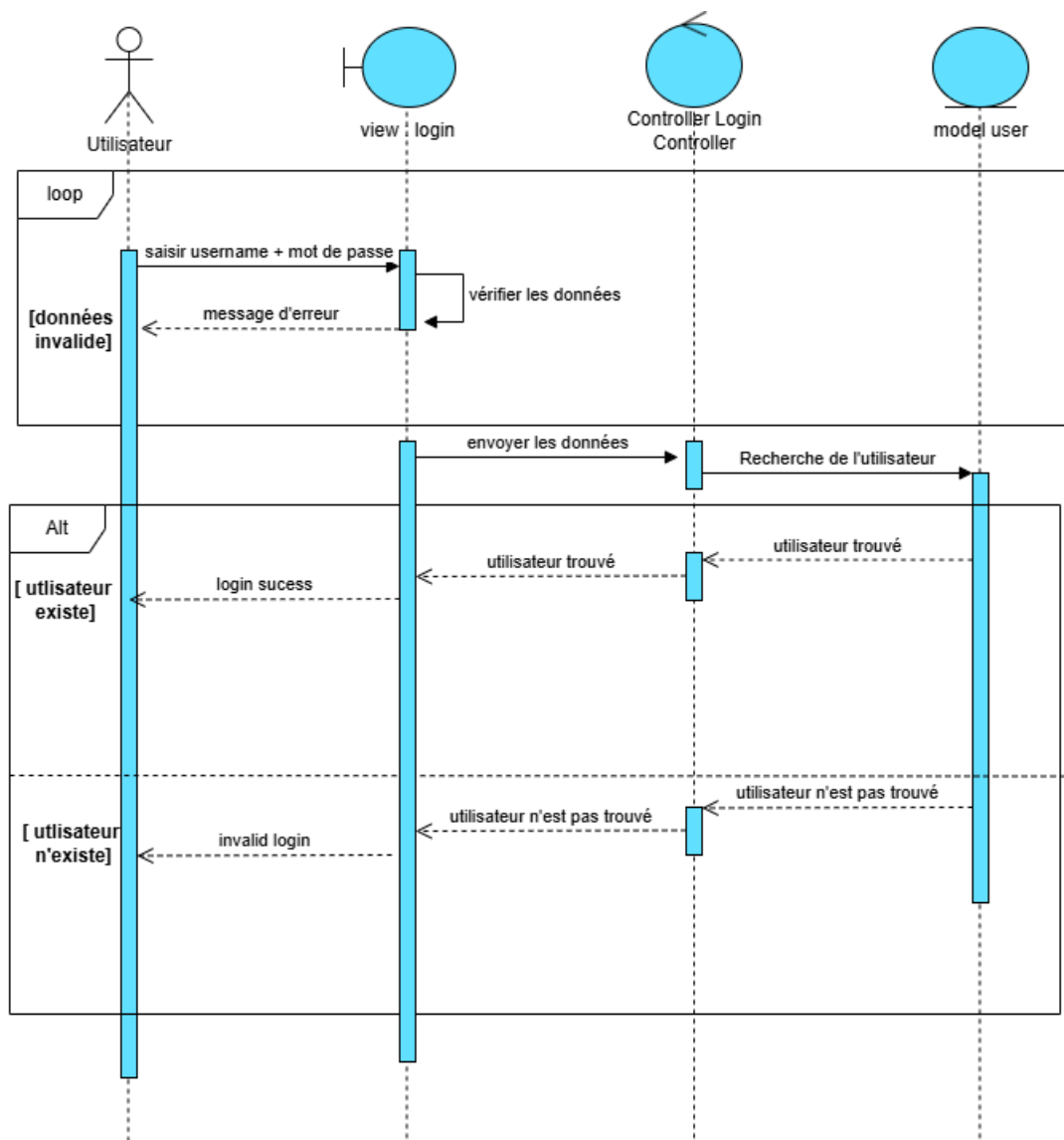


Figure 6.3 – Diagramme de séquence pour l'authentification

○ Diagramme de séquence détaillée(Supprimer une vente)

Ce diagramme de séquence UML illustre le processus de suppression d'une vente par l'Administrateur, mettant en évidence les interactions entre divers composants du système. L'Administrateur commence par sélectionner une vente à supprimer depuis l'interface du système. Suite à la sélection de la vente, le système envoie une demande de confirmation au VenteController, qui affiche cette demande à l'Administrateur pour confirmer son intention de supprimer la vente. Si l'Administrateur confirme la suppression, le contrôleur procède à la suppression en communiquant avec la base de données et supprime les images dans le dossier. Si l'Administrateur choisit d'annuler, aucune action n'est exécutée. Dans les deux situations, une nouvelle liste des ventes est générée et affichée à l'Administrateur.

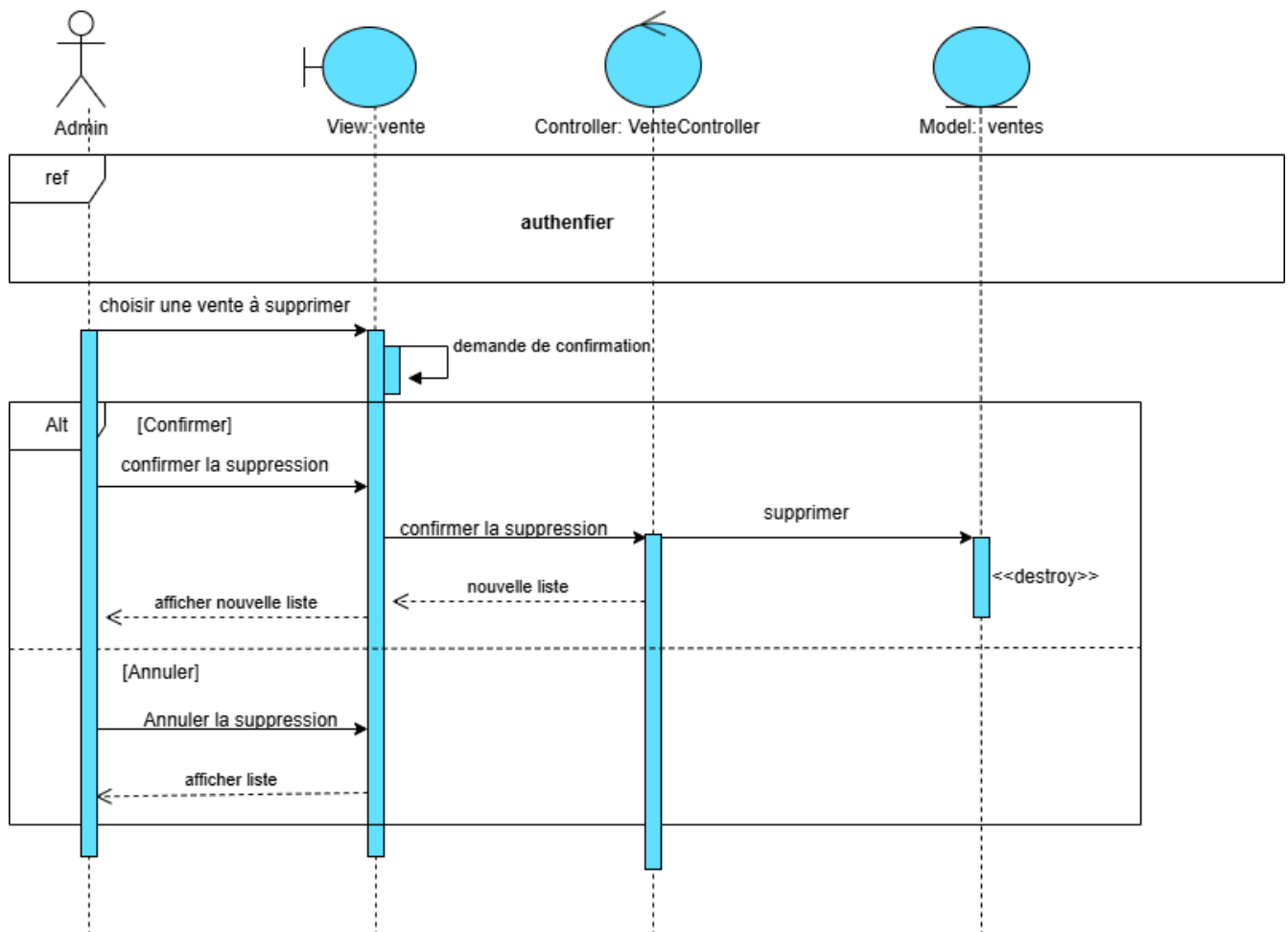


Figure 6.4 – Diagramme de séquence pour la suppression d'une vente

○ **Diagramme de séquence détaillée(ajouter un utilisateur)**

Ce diagramme de séquence illustre le processus d'ajout d'un nouvel utilisateur par un administrateur dans une application. L'administrateur saisit les données et télécharge des images via l'interface d'inscription. Le contrôleur de connexion vérifie les données et, si elles sont valides, enregistre les images et les informations de l'utilisateur. Les images sont stockées dans le système de fichiers et les données dans la base de données des utilisateurs. La nouvelle liste des utilisateurs est ensuite retournée et affichée à l'administrateur. Si les données sont invalides, un message d'erreur est renvoyé, demandant à l'administrateur de corriger les informations saisies. Ce processus garantit l'intégrité des données et la mise à jour correcte de la liste des utilisateurs.

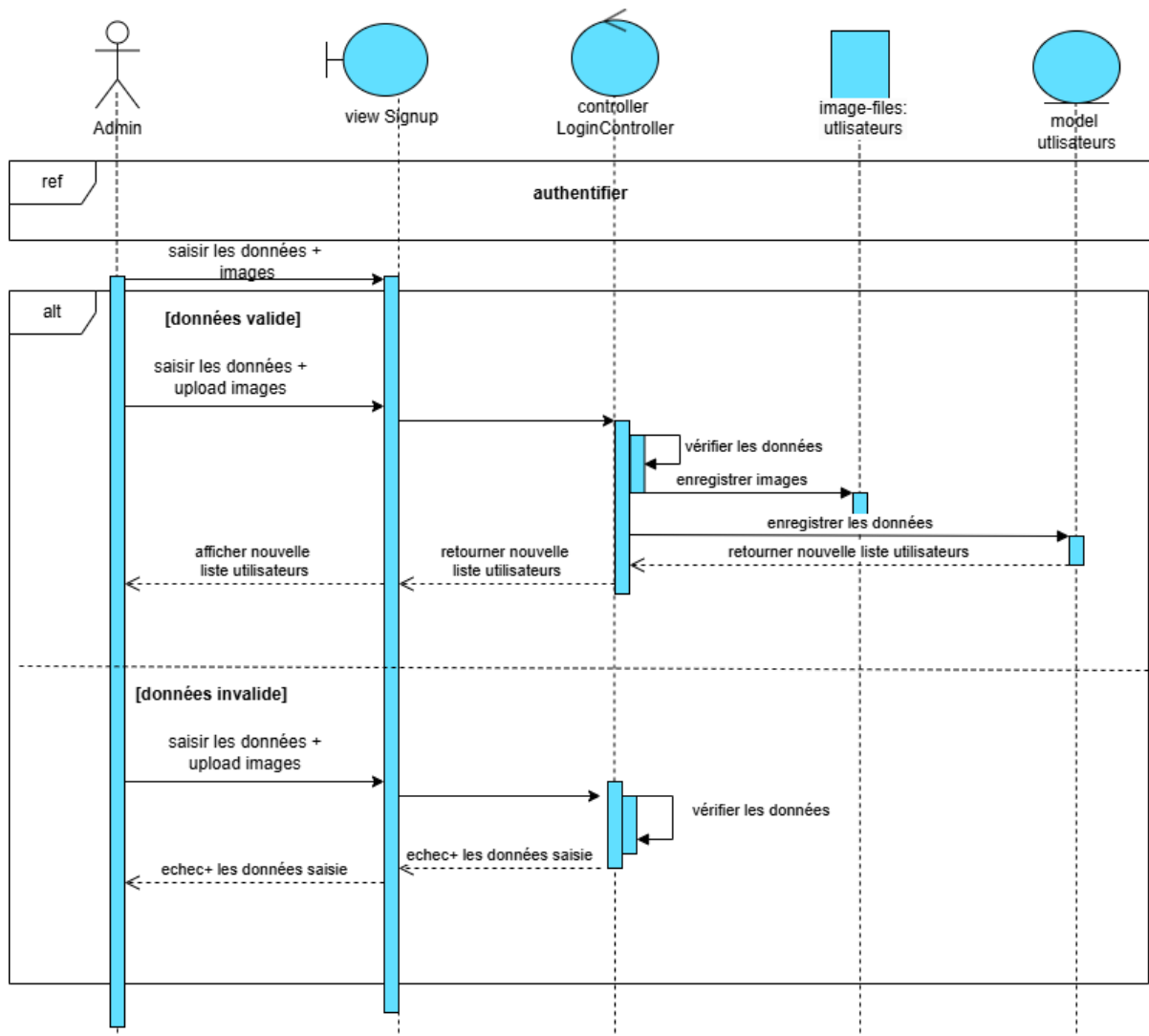


Figure 6.5 – Diagramme de séquence pour ajouter utilisateur

6.2.3 Réalisation du Sprint 4

Les figures ci-dessous illustrent des exemples d'interfaces graphiques développées au cours de ce sprint.

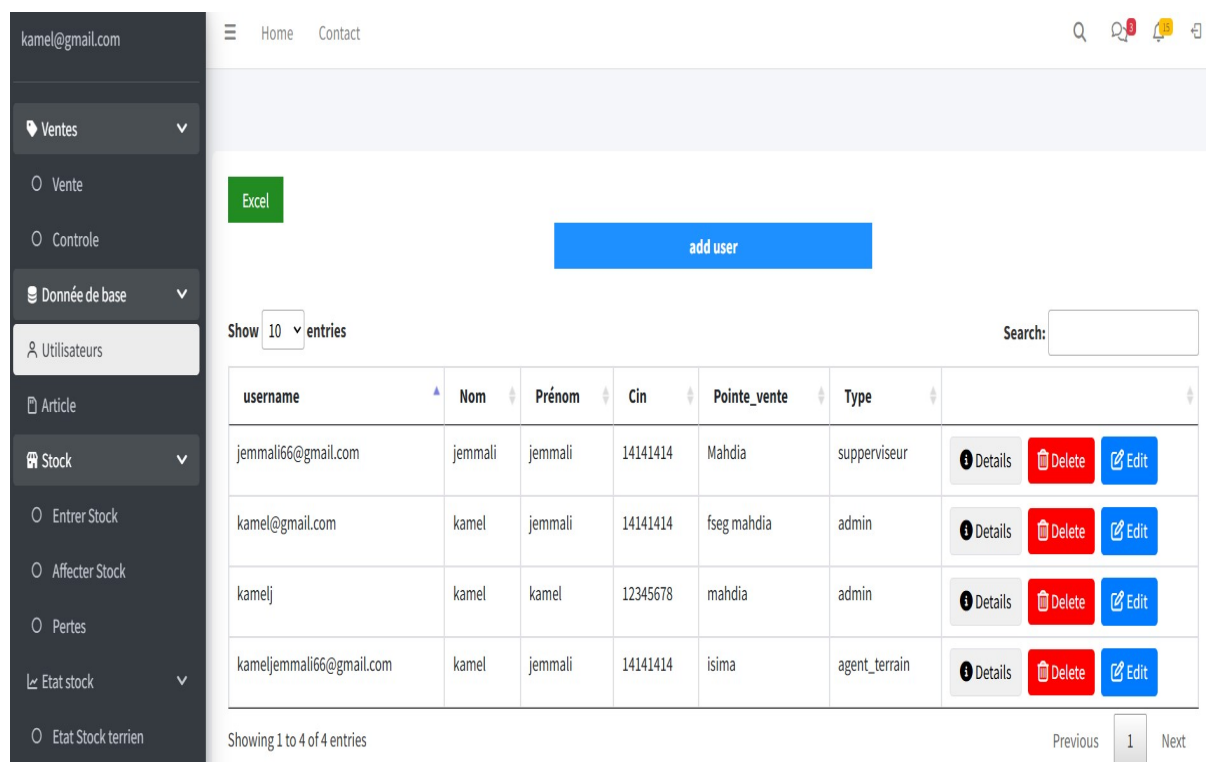


Figure 6.6 – Interface pour la liste des utilisateurs

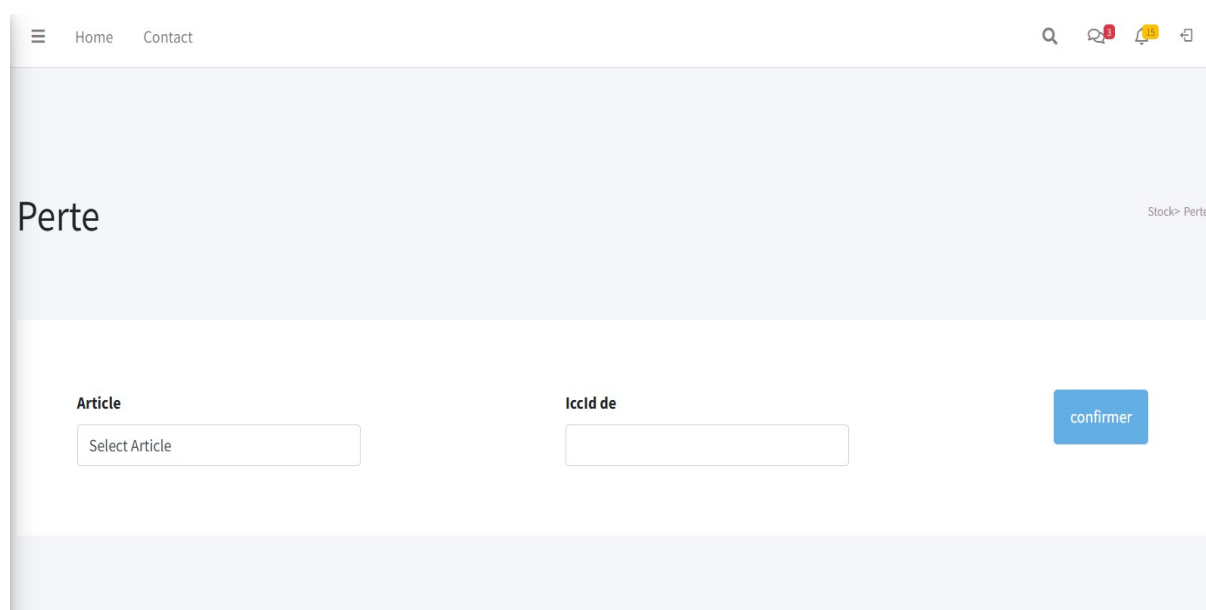


Figure 6.7 – Interface de dceclaration des pertes

6.3 Conclusion

Dans ce chapitre, nous avons exploré en détail les rôles et les responsabilités de l'administrateur au sein de notre système. À travers des diagrammes de cas d'utilisation, des diagrammes de classes et des diagrammes de séquence, nous avons mis en lumière comment l'administrateur gère les utilisateurs, les permissions et les configurations globales. Ces éléments assurent le bon fonctionnement et la sécurité de notre système, garantissant que toutes les opérations sont effectuées de manière fluide et sécurisée. L'administrateur joue un rôle central dans la gestion de l'infrastructure du système, ce qui est crucial pour maintenir une plateforme stable et fiable.

Conclusion générale

Ce document a présenté en détail une étape cruciale de notre projet de fin d'études, qui consiste en la conception et le développement d'un système de gestion des ventes de puces téléphoniques. Nous avons commencé par introduire l'entreprise avec laquelle nous avons collaboré, en mettant en lumière les objectifs et les défis auxquels elle fait face. Nous avons ensuite identifié les problèmes existants et analysé les solutions actuelles.

La majeure partie de ce rapport a été consacrée à la description de la conception et de l'architecture de notre solution, ainsi qu'à l'exécution du travail planifié, qui s'est déroulé selon la méthodologie Scrum. Le système que nous avons développé intègre des fonctionnalités avancées telles que la gestion en temps réel des ventes et des stocks, le pointage avec reconnaissance faciale, et la géolocalisation des agents de terrain. Ces innovations visent à améliorer l'efficacité opérationnelle et à offrir une expérience utilisateur optimisée.

Notre projet a permis d'implémenter un système où les agents peuvent effectuer des ventes, enregistrer des pointages et signaler des pertes de manière fluide et efficace. Les administrateurs disposent d'outils robustes pour gérer les utilisateurs, vérifier les ventes et surveiller les opérations globales. Grâce à cette solution, nous avons réussi à créer une interface utilisateur intuitive et des processus automatisés qui simplifient les tâches répétitives et minimisent les erreurs.

Cette expérience de développement nous a offert une occasion inestimable de plonger profondément dans les détails techniques et pratiques du développement web et mobile. Nous avons appris à naviguer dans les complexités des frameworks modernes et à intégrer diverses technologies pour créer une solution cohérente et performante.

En conclusion, ce projet nous a non seulement permis d'acquérir de précieuses compétences

techniques, mais il nous a aussi montré l'importance de la collaboration, de la planification rigoureuse et de l'adaptabilité dans le monde du développement logiciel. Nous réalisons que, malgré tout ce que nous avons appris, il reste encore beaucoup à découvrir et à maîtriser. Ce projet a été un véritable voyage d'apprentissage, renforçant notre passion pour le développement et notre détermination à continuer d'explorer et d'innover dans ce domaine dynamique.

PERSPECTIVES

Alors que notre système a été récemment mis en place, il est essentiel de débiter avec un nombre limité d'utilisateurs pour évaluer les scénarios réels et recueillir des retours qui seront cruciaux pour améliorer continuellement les fonctionnalités et l'expérience utilisateur. Il y a encore un chemin important à parcourir pour que l'application atteigne les standards que nous visons à long terme.

Sur le plan technique, nous envisageons de transformer les vues rendues côté serveur en une application web spécifiquement dédiée à l'administration de la plateforme. Le développement de l'interface utilisateur serait plus pratique et gérable en utilisant un framework moderne côté client. Pour l'application serveur, l'utilisation de gRPC pour les microservices est à l'étude, bien que cette technologie soit relativement nouvelle et que nous soyons encore en phase d'évaluation.

Une des fonctionnalités critiques à développer pour l'application mobile est l'intégration d'un système de chat interne. Ce système de messagerie permettra aux agents de terrain de communiquer efficacement entre eux en temps réel, facilitant ainsi la coordination et l'échange d'informations pertinentes sur le terrain. Cette fonctionnalité vise à renforcer la collaboration et à améliorer la gestion des opérations en direct.

En plus de ces développements, nous introduisons un système de liste noire pour empêcher les abus dans l'utilisation des SIM. Si un client dépasse le seuil défini par l'entreprise, il sera automatiquement ajouté à la liste noire et bloqué pour les transactions futures. Les agents recevront un message d'erreur lors des tentatives de transaction avec une personne listée, assurant ainsi que nos services ne sont pas exploités de manière frauduleuse.

Ces améliorations et ajouts sont conçus pour améliorer l'efficacité opérationnelle et renforcer la sécurité, tout en offrant une expérience utilisateur optimisée et un environnement de travail collaboratif pour nos agents.

