

République Algérienne Démocratique et Populaire
Ministère de L'Enseignement Supérieur et de la Recherche Scientifique
Université Abderrahmane MIRA – Béjaïa

Faculté Des Sciences Exactes
Département d'Informatique



Mémoire de fin de cycle

En vue de l'obtention du diplôme de Master En Informatique
Projet pour l'obtention d'un diplôme-start-up dans le cadre de
l'arrêté ministériel n°1275

Option :Génie Logiciel

T H È M E

**Conception et réalisation d'une application mobile
"Wasla" pour le transport de marchandises**

Réalisé par :

M. BOUYAHI Lounes M. SOUYET Zoubir

Devant le jury composé de :

Président : M. Djoudi TOUAZI, M.C.B, U.A.M Béjaïa

Examineur : M. Fatah BOUCHEBBAH, M.C.B, U.A.M Béjaïa

Encadrant : M. Mohamed MOHAMMEDI, M.C.A, U.A.M Béjaïa

Année universitaire : 2022/2023

✧ Remerciements ✧

Nous tenons tout d'abord à remercier Dieu le tout puissant qui nous a donné la force et la patience d'accomplir ce modeste travail.

*Nous tenons à remercier notre encadrant, Monsieur **MOHAMMEDI Mohamed**, pour son précieux soutien, ses conseils éclairés et sa patience tout au long de ce travail. Ses orientations et ses suggestions ont grandement enrichi notre travail.*

Nous tenons également à remercier les membres du jury qui nous font le grand honneur d'évaluer ce travail.

*Un grand merci à notre ami Monsieur **AISSAOUI Ahmed** pour le partage de son expérience dans le domaine avec nous*

À nos chers parents, qui nous ont toujours encouragés et soutenus durant toute la période de nos études jusqu'à atteindre ce stade de notre formation en général et notre projet de fin de cycle en particulier.

Je tiens à exprimer ma reconnaissance envers mes amis et ma famille pour leur soutien inconditionnel et leurs encouragements tout au long de cette période d'études. Leur soutien moral et leur compréhension ont été essentiels pour surmonter les défis.

Enfin, je tiens à remercier l'ensemble de l'équipe enseignante de l'Université Abderrahmane MIRA – Béjaïa pour la qualité de leur enseignement et pour avoir façonné ma compréhension du sujet.

** Dédicaces **

Je dédie ce modeste travail aux personnes chères à mon cœur.

À mes parents, pour leur soutien et leur amour.

À mes quatre frères.

À mes cousins.

À mes amis.

À mes copains de résidence : K101 R.U 17 octobre, J102 R.U 17 octobre.

Et à toutes les personnes qui m'ont apporté de l'aide

Zoubir

✧ *Dédicaces* ✧

Je dédie ce travail aux personnes chères à mon cœur. À mes parents, pour leur soutien et leur amour. À mes frères et sœurs. À mes cousins. À mes amis. À mes copains de résidence. Et à toutes les personnes qui m'ont apporté de l'aide. Votre présence et votre encouragement ont été inestimables tout au long de ce parcours académique, et c'est avec gratitude que je vous offre ce travail.

Lounes

TABLE DES MATIÈRES

Table des Matières	i
Liste des tableaux	v
Liste des figures	vi
Liste des acronymes	viii
Introduction générale	1
1 Présentation de l'organisme d'accueil et méthodologie de conception	3
1.1 Introduction	3
1.2 Contexte et justification de l'étude	3
1.3 Problématique et objectifs	4
1.4 Hypothèses	4
1.5 Présentation de l'idée de startup	5
1.5.1 Concept de l'application	5
1.5.2 Vision et mission de la startup	5
1.5.3 Analyse du marché cible	5
1.6 Méthodologie de développement	6
1.6.1 Présentation de la méthode scrum	6
1.6.2 Acteurs de la méthode Scrum	7
1.6.3 Product Backlog	7
1.6.4 Users stories	7
1.6.5 Les piliers de la méthode	8
1.7 Conclusion	8

2	Sprint zéro – Préparation et Planification Scrum	9
2.1	Introduction	9
2.2	Equipe et rôles	9
2.3	Spécification des besoins	10
2.3.1	Besoins fonctionnels	10
2.3.2	Besoins non fonctionnels	11
2.4	Modalisation des besoins	11
2.4.1	Identification des acteurs du system	11
2.4.2	Digramme de cas d'utilisation global	12
2.5	Backlog du produit	13
2.6	Planification des sprints	14
2.7	Conclusion	14
3	Conception	16
3.1	Introduction	16
3.2	Étude du premier sprint	16
3.2.1	User story	16
3.2.2	Décomposition du sprint 1	17
3.2.3	Diagramme de cas d'utilisation	18
3.2.4	Cas d'utilisation « Inscription client »	18
	Description textuelle	18
	Diagramme de séquence détaillé	20
3.2.5	Cas d'utilisation « Inscription transporteur»	20
	Description textuelle	20
	Diagramme de séquence détaillé	22
3.2.6	Cas d'utilisation « S'authentifier »	23
	Description textuelle	23
	Diagramme de séquence détaillé	24
3.2.7	Diagramme de classes	25
3.2.8	Modèle relationnel	25
3.2.9	Maquette	26
	Interfaces d'Inscription	26
	Interfaces d'Authentification	27
	L'interface d'administrateur	28
3.3	Étude du deuxième sprint	29
3.3.1	User story	29

3.3.2	Décomposition du sprint 2	30
3.3.3	Diagramme de cas d'utilisation	30
3.3.4	Cas d'utilisation «Demander une course»	31
	Description textuelle	31
	Diagramme de séquence détaillé	33
3.3.5	Diagramme de classe	34
3.3.6	Modèle relationnel	34
3.3.7	Maquette	35
	Interface de demande de Course	35
3.3.8	Gestion des Courses et Acceptation	35
3.4	Étude du Troisième sprint	36
3.4.1	User story	36
3.4.2	Décomposition du sprint 3	37
3.4.3	Diagramme de cas d'utilisation	38
3.4.4	Cas d'utilisation «Suivre état de course»	39
	Description textuelle	39
	Diagramme de séquence détaillé	40
3.4.5	Diagramme de classe	41
3.4.6	Modèle relationnel	41
3.4.7	Maquette	42
3.5	Conclusion	45
4	Réalisation	46
4.1	Introduction	46
4.2	Environnement et outils de développement	46
4.2.1	Visual Paradigm	46
4.2.2	Draw.io	47
4.2.3	Lucidchart	47
4.2.4	Visual Studio	47
4.2.5	Android Studio	48
4.2.6	Expo	48
4.3	Front-end	49
4.3.1	React Native	49
4.4	Back-end	49
4.4.1	Node.js	49
4.5	Implémentation de la base de données	50

4.5.1	NoSQL	50
4.5.2	Fairebase	50
4.6	Conclusion	50
Conclusion générale et perspectives		52
Bibliographie		54

LISTE DES TABLEAUX

2.1	Identification des acteurs du système.	10
2.2	Identification des acteurs du système.	11
2.3	Présentation backlog du produit.	14
2.4	Planification des sprints.	14
3.1	Description textuelle du cas d'utilisation «Inscription client»	19
3.2	Description textuelle du cas d'utilisation "Inscription transporteur"	22
3.3	Description textuelle du cas d'utilisation "Authentification"	23
3.4	Description textuelle du cas d'utilisation "Demander une course"	32
3.5	Description textuelle du cas d'utilisation "Suivre état de course"	39

TABLE DES FIGURES

1.1	Cycle de vie de la méthode Scrum.	6
2.1	Diagramme de cas d'utilisation global.	12
3.1	User Story du sprint 1.	17
3.2	Décomposition du sprint 1 en termes de temps.	17
3.3	Diagramme de cas d'utilisation du sprint 1.	18
3.4	Diagramme de séquence détaillé du cas «Inscription client»	20
3.5	Diagramme de séquence détaillé du cas «Inscription transporteur».	22
3.6	Diagramme de séquence détaillé du cas « S'authentifier ».	24
3.7	Diagramme de class du sprint 1.	25
3.8	Interface « Inscription client ».	26
3.9	Interface « Inscription transporteur ».	26
3.10	Interface « Authentification client ».	27
3.11	Interface « Authentification transporteur ».	27
3.12	Interface «valider transporteur ».	28
3.13	Interface « suspendre transporteur ».	28
3.14	User Story du sprint 2.	29
3.15	Décomposition du sprint 2 en termes de temps.	30
3.16	Diagramme de cas d'utilisation du sprint 2.	31
3.17	Diagramme de séquence détaillé du cas «Demander une course».	33
3.18	Diagramme de classe du sprint 2.	34
3.19	Interface « Demande course ».	35
3.20	Interface « Demande de course ».	35
3.21	Interface « Liste des course coté transporteur ».	36

3.22	Dnterface « Detaille de course »	36
3.23	User Story du sprint 3.	37
3.24	User Story du sprint 3.	37
3.25	Diagramme de cas d'utilisation du sprint 3.	38
3.26	Diagramme de séquence détaillé du cas «Suivre état de course».	40
3.27	Diagramme de classe du sprint 3.	41
3.28	Interface « liste dse courses ».	42
3.29	Interface « modifier l'état d'une course ».	43
3.30	Interface « tableau de bord de l'administrateur ».	44
4.1	Logo Visual Paradigm.	46
4.2	Logo Draw.io.	47
4.3	Logo Lucidchart.	47
4.4	Logo Visual Studio.	48
4.5	Logo Android Studio.	48
4.6	Logo Expo.	49
4.7	Logo React Native.	49
4.8	Logo Fairebase.	50

LISTE DES ACRONYMES

A	API	Application programming interface.
E	EDI	environnement de développement intégr.
	ID	User story.
I	IOS	iPhone Operating System
	ID	User story.
S	SQL	Structured Query Language
	U.S	User story.
U	UML	Unified Modeling Language.
W	WEB	World Wide Web
	NoSQL	: Not only SQL.

INTRODUCTION GÉNÉRALE

Le secteur du transport de marchandises est au cœur de l'économie mondiale. Il joue un rôle essentiel dans la distribution de biens et services, et il contribue à la croissance économique [15]. Cependant, ce secteur est souvent confronté à des défis, tels que l'inefficacité, le manque de transparence et les coûts élevés.

En Algérie, le secteur du transport de marchandises est en pleine croissance. L'économie algérienne se développe, et la demande de transport de marchandises augmente. Cependant, le secteur est également confronté à des défis spécifiques, tels que l'infrastructure limitée et la bureaucratie [8].

Dans ce contexte, notre mémoire de fin de cycle se penche sur le développement d'une solution innovante pour répondre aux besoins du marché algérien du transport de marchandises. Cette solution est une application mobile dédiée à la demande de transport de marchandises.

L'application mobile "Wasla" a pour objectif de simplifier et améliorer le processus de demande de transport de marchandises pour les entreprises, les expéditeurs et les transporteurs en Algérie. Elle offrira une interface conviviale et transparente, permettant aux utilisateurs de soumettre leurs demandes en quelques clics.

Le développement de notre application "Wasla" repose sur la méthode Scrum, une méthodologie agile de gestion de projet qui favorise la flexibilité, la collaboration et l'itération. Cette méthodologie nous a permis de développer une application qui répond aux besoins des utilisateurs et qui est adaptable aux changements.

Notre mémoire de fin de cycle est structuré en quatre chapitres. Dans le premier chapitre, nous avons présenté l'organisme d'accueil, détaillé notre méthodologie de conception, exposé le contexte, la problématique, les objectifs et l'idée de startup associée à notre application de transport de marchandises. Nous avons également introduit la méthodologie de développement, mettant en avant l'utilisation de la méthode Scrum.

Le deuxième chapitre, est consacré au "Sprint zéro" de notre projet, avec un focus sur

l'équipe, la spécification des besoins, la modélisation des besoins, le backlog du produit et la planification des sprints.

Le troisième chapitre, est dédié a la phase de conception de notre application, détaillant chaque sprint, les user stories, les diagrammes de cas d'utilisation, les diagrammes de classe, les modèles relationnels et les maquettes.

Dans le quatrième chapitre, nous avons présenté la phase de réalisation de l'application. Nous avons décrit l'environnement de développement, les outils utilisés, le front-end, le back-end et l'implémentation de la base de données

Cette structure méthodologique nous a permis d'aborder de manière exhaustive chaque étape de notre projet d'application mobile pour le transport de marchandises, garantissant ainsi une approche complète et structurée de notre travail de recherche et de développement.

Enfin ce mémoire s'achève par une conclusion générale et perspective que nous souhaitons accomplir prochainement.

CHAPITRE 1

PRÉSENTATION DE L'ORGANISME D'ACCUEIL ET MÉTHODOLOGIE DE CONCEPTION

1.1 Introduction

Le secteur du transport de marchandises connaît une évolution significative en Algérie, en parallèle avec l'augmentation de l'utilisation des smartphones et le développement du commerce électronique. Les particuliers et les entreprises algériens expriment un besoin croissant envers un service de transport de marchandises efficace, rapide et fiable. C'est dans ce contexte que notre startup a choisi de se lancer dans le développement de l'application mobile "Wasla" dédiée à la demande de transport de marchandises en Algérie. L'objectif principal de notre application est de simplifier le processus de transport de marchandises en facilitant la mise en relation entre les propriétaires de véhicules de transport et les clients, tout en améliorant la visibilité sur le suivi des livraisons et en favorisant une communication fluide entre toutes les parties impliquées.

Ce chapitre introductif présente notre startup, l'importance du projet, ainsi que les défis, objectifs et spécificités liés à son lancement en Algérie.

1.2 Contexte et justification de l'étude

Le marché du transport de marchandises en Algérie est en pleine mutation, notamment grâce à l'adoption croissante des smartphones et à l'évolution du comportement d'achat des consommateurs. Les besoins de transport de marchandises, que ce soient des biens personnels ou des produits commerciaux, sont de plus en plus pressants. Dans ce contexte, notre startup a pris l'initiative de concevoir une application mobile dédiée à la demande de transport de marchandises, adaptée aux spécificités algériennes.

1.3 Problématique et objectifs

Le transport de marchandises en Algérie fait face à de nombreux défis. Pour les clients, la recherche d'un transporteur fiable, proposant des tarifs compétitifs et respectant les délais de livraison, est un véritable casse-tête. De leur côté, les propriétaires de véhicules de transport cherchent des moyens pour obtenir des clients réguliers et rentables. Toutefois, le principal obstacle dans ce secteur réside dans la communication insuffisante et le manque de confiance mutuelle entre les parties impliquées.

Dans ce contexte, notre startup s'est fixée pour mission de résoudre ces problèmes en proposant une solution innovante, l'application mobile "Wasla". Cette application vise à faciliter la liaison entre les clients et les transporteurs en offrant une plateforme transparente et sécurisée, ce qui améliorera l'efficacité du transport de marchandises et contribuera au développement économique en Algérie.

Les questions fondamentales auxquelles nous nous attaquerons dans cette étude sont les suivantes :

- Comment répondre de manière efficace aux problèmes rencontrés par les clients et les propriétaires de véhicules de transport en Algérie dans le domaine de la logistique de marchandises grâce à une application mobile ?
- Quelles fonctionnalités essentielles l'application doit-elle offrir pour améliorer l'expérience utilisateur et optimiser les opérations logistiques en Algérie ?
- Comment concevoir une application mobile conviviale et adaptée aux besoins des différents acteurs de la chaîne logistique en Algérie ?
- Comment mettre en place une méthodologie de conception appropriée pour développer une application mobile de qualité et fiable en Algérie ?

Notre application vise à simplifier le processus de transport de marchandises en mettant en relation les propriétaires de véhicules de transport et les clients en Algérie. Elle vise également à améliorer la réservation et la gestion des transports, en offrant une visibilité accrue sur l'état des livraisons et en favorisant une communication fluide entre toutes les parties impliquées.

1.4 Hypothèses

- Nous supposons que notre application mobile de demande de transport de marchandises, adaptée au marché algérien, peut potentiellement répondre à un besoin non comblé et susciter un intérêt considérable parmi les utilisateurs locaux.
- Nous présumons que la sécurité, la fiabilité et la commodité du transport de marchandises

sont des éléments cruciaux pour les utilisateurs algériens, et nous cherchons à déterminer comment notre application peut répondre à ces préoccupations.

- Nous envisageons que l'attrait des avantages économiques et de la visibilité accrue pourrait inciter les transporteurs en Algérie à participer à notre plateforme, mais nous souhaitons comprendre les incitations spécifiques qui les encourageraient à s'engager.

1.5 Présentation de l'idée de startup

1.5.1 Concept de l'application

Notre startup se concentre sur la création d'une application mobile dédiée à la demande de transport de marchandises en Algérie. L'application a pour mission de simplifier le transport de marchandises pour les particuliers et les entreprises en Algérie. Cette application permettra aux utilisateurs de soumettre facilement une demande de transport de marchandises en quelques clics. Les utilisateurs auront la possibilité de spécifier les détails de leur demande, tels que le type de marchandises, le point de départ et la destination, ainsi que la date et l'heure souhaitées pour le transport. Les transporteurs pourront également utiliser l'application pour trouver des demandes de transport à proximité.

L'application mobile "Wasla" pour le transport de marchandises en Algérie est conçue pour offrir une expérience utilisateur intuitive, simple et rapide. Nous souhaitons fournir un service efficace et fiable à tous les utilisateurs, qu'ils soient particuliers ou entreprises.

1.5.2 Vision et mission de la startup

Notre startup a pour vision de révolutionner le marché du transport de marchandises en Algérie en proposant un service simple, rapide et fiable. Nous aspirons à devenir la référence en matière de demande de transport de marchandises. Notre mission est de simplifier le processus de demande de transport de marchandises en utilisant la technologie mobile pour mettre en relation les demandeurs et les transporteurs de manière efficace et transparente.

1.5.3 Analyse du marché cible

Notre startup se concentre sur les particuliers et les entreprises en Algérie qui ont besoin de transporter leurs marchandises. Nous avons identifié que les particuliers peuvent avoir besoin de transporter des objets volumineux, tels que des meubles ou des appareils électroménagers. Les entreprises peuvent également avoir besoin de transporter des marchandises pour leurs activités, telles que des matériaux de construction ou des produits finis. Nous avons également constaté

que le marché de la demande de transport de marchandises en Algérie est en constante évolution, avec l'augmentation de l'utilisation des smartphones et l'émergence du commerce électronique. Notre application mobile répondra à ces besoins en offrant un service simple, rapide et fiable pour tous les utilisateurs en Algérie.

1.6 Méthodologie de développement

La création d'une application informatique implique un processus comprenant plusieurs étapes, conformément à des approches de développement telles que les méthodologies agiles. Les méthodes agiles se concentrent sur une approche de gestion des projets informatiques qui repose sur un dialogue entre toutes les parties prenantes : développeurs, clients et autres professionnels du projet. Elles mettent également l'accent sur la rapidité de livraison, la flexibilité pour ajuster les plans en cours de réalisation et la capacité d'adaptation. Afin de répondre aux exigences du client tout en optimisant les coûts et les délais, nous adopterons la méthode agile "SCRUM" [1].

1.6.1 Présentation de la méthode scrum

La méthode SCRUM est un processus de développement qui est basé sur une coordination entre l'équipe de développement, la direction de l'entreprise et le client. Il exige d'appliquer des méthodes de travail et de communication qui sont nouvelles pour le bien des individus. Elle requiert du client de participer dans le projet [1].

La Figure 1.1 présente les composants de la méthode Scrum.

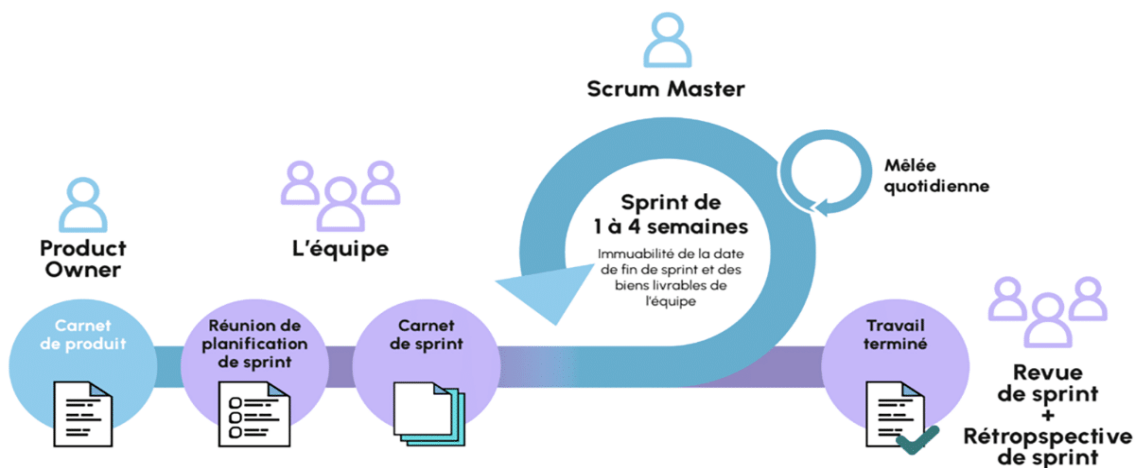


FIGURE 1.1 – Cycle de vie de la méthode Scrum.

1.6.2 Acteurs de la méthode Scrum

Product Owner : assume le rôle de porte-parole des clients (utilisateurs) au sein du projet. Doté d'une compréhension approfondie du produit, il est en mesure d'anticiper les évolutions souhaitées et élabore une stratégie pour garantir l'adéquation du produit aux besoins des clients. De plus, il est chargé de maintenir à jour le carnet de produits (product backlog) [2].

L'équipe de développement : est chargée de fournir, à la conclusion de chaque Sprint, les fonctionnalités développées qui ont été sélectionnées en tant que priorités pour ce Sprint. Composée de deux à dix membres, cette équipe assume également la tâche d'estimer les histoires d'utilisateurs [2].

Scrum Master : membre de l'équipe, il a pour objectif d'améliorer la capacité de production de l'équipe. Son rôle principal est la mise en œuvre de la méthode Scrum qu'il maîtrise parfaitement [2].

1.6.3 Product Backlog

Le Product Backlog est une liste ordonnée et émergente des éléments nécessaires pour améliorer le produit, et il sert de seule source de travail pour l'équipe Scrum.

Les éléments du Product Backlog qui peuvent être complétés au cours d'un seul Sprint sont prêts à être inclus dans la planification du Sprint après avoir été suffisamment raffinés. Le processus de raffinement du Product Backlog consiste à décomposer et à préciser davantage les éléments en éléments plus petits et plus détaillés. Cette activité est continue et vise à ajouter des détails tels qu'une description, un ordre de priorité et une estimation de la taille. Ces attributs spécifiques peuvent varier en fonction du domaine d'activité.

Les développeurs chargés de la réalisation du travail sont responsables de cette estimation, mais le Product Owner peut les influencer en clarifiant ses exigences et en les aidant à trouver des compromis lorsque cela est nécessaire [3].

1.6.4 Users stories

Les fonctionnalités décrites portent le nom de User Stories et sont décrites en employant la terminologie utilisée par le client [5]. Dans notre cas, nous avons divisé chaque user story en trois(03) tâches qui sont les suivantes :

- Réaliser les diagrammes UML, les descriptions textuelles et le modèle relationnel de la fonctionnalité.
- Développer la fonctionnalité .
- Tester la fonctionnalité.

1.6.5 Les piliers de la méthode

La méthode Scrum repose sur trois piliers fondamentaux qui constituent sa base de fonctionnement.

- **Transparence** Ce pilier implique que toutes les informations pertinentes concernant le projet, le travail en cours, les défis et les progrès doivent être visibles et accessibles à tous les membres de l'équipe Scrum. La transparence favorise la confiance, la communication ouverte et la prise de décision éclairée.
- **Inspection** L'inspection consiste à examiner régulièrement les artefacts du projet, tels que le produit en cours de développement, les processus de travail et les résultats des Sprints. Cette évaluation continue permet de détecter les problèmes, les erreurs et les opportunités d'amélioration.
- **Adaptation** Sur la base des informations collectées lors des inspections, l'équipe Scrum doit être prête à apporter des ajustements et des améliorations constantes. L'adaptation signifie que l'équipe doit être flexible et capable de réagir rapidement aux changements, aux nouvelles priorités et aux retours d'expérience [4].

1.7 Conclusion

Ce chapitre a présenté les bases de notre projet de startup, axé sur une application mobile dédiée au transport de marchandises. Nous avons présenté le contexte, les défis à relever, les objectifs de recherche, ainsi que la méthodologie de développement, en mettant en avant la méthode Scrum. Dans le prochain chapitre, nous nous concentrerons sur le sprint zéro, une étape importante dans le processus de développement Scrum.

CHAPITRE 2

SPRINT ZÉRO – PRÉPARATION ET PLANIFICATION

SCRUM

2.1 Introduction

Dans la chronologie du développement de notre projet, le sprint zéro représente une étape cruciale pour établir une base solide et préparer les éléments essentiels en vue du lancement du Sprint 1. Au sein de ce chapitre, nous entamons notre plongée dans la méthodologie Scrum en abordant la planification des sprints. Nous débuterons par la présentation de l'équipe Scrum et la spécification des besoins fonctionnels et non fonctionnels. Par la suite, nous explorerons la création du Product Backlog, la priorisation et l'estimation des sprints, et nous conclurons par la planification de ces derniers, définissant ainsi une feuille de route claire pour la réalisation de notre projet de développement d'une application de transport de marchandises.

2.2 Equipe et rôles

Pour notre projet, les rôles sont repartis comme indiqué dans le Tableau 2.1 suivant :

Rôle Scrum	Personne affectée
Product owner	M. SOUYET zoubir M. BOUYAHI lounes
Scrum master	M. Mohamed MOHAMMEDI
Équipe Scrum	M. SOUYET zoubir M. BOUYAHI Lounes

TABLEAU 2.1 – Identification des acteurs du système.

2.3 Spécification des besoins

2.3.1 Besoins fonctionnels

Les besoins fonctionnels correspondent aux fonctionnalités que l'application doit fournir aux utilisateurs. Ils sont décrits ci-dessous :

- **Inscription** : les utilisateurs (transporteurs) doivent pouvoir s'inscrire sur l'application pour créer un compte.
- **Authentification** : les utilisateurs doivent pouvoir se connecter à l'application avec leurs identifiants pour accéder à leur compte.
- **Demande de course** : les clients doivent pouvoir demander une course pour transporter une marchandise en temps réel en spécifiant les détails de la livraison souhaitée (point de départ, point d'arrivée, type de marchandise, etc.).
- **Acceptation de course** : les propriétaires de véhicules (transporteur) doivent pouvoir accepter ou refuser les courses proposées par le système.
- **Communication client-propriétaire de véhicule** les propriétaires de véhicules doivent pouvoir communiquer avec les clients pour organiser les détails de la livraison (horaire, lieu de rendez-vous, etc.).
- **Paiement** : les clients doivent pouvoir effectuer le paiement de la course directement via l'application.
- **Historique de course** : les utilisateurs doivent pouvoir consulter l'historique de leurs courses passées.
- **Notification** : les utilisateurs doivent pouvoir recevoir des notifications sur l'état de leur course en temps réel (confirmation de commande, acceptation de course, confirmation de

livraison, etc.).

2.3.2 Besoins non fonctionnels

Les besoins non fonctionnels correspondent aux contraintes que l'application doit respecter. Ils sont décrits ci-dessous :

- **Performance** : l'application doit être rapide et réactive pour offrir une expérience utilisateur fluide.
- **Sécurité** : l'application doit garantir la sécurité des données des utilisateurs et des transactions financières effectuées sur la plateforme.
- **Ergonomie** : l'interface utilisateur de l'application doit être ergonomique et facile à utiliser pour permettre aux utilisateurs de naviguer aisément sur la plateforme.
- **Disponibilité** : le système doit être disponible 24h/24 et 7j/7 pour permettre aux utilisateurs d'accéder à la plateforme à tout moment.

2.4 Modalisation des besoins

2.4.1 Identification des acteurs du system

Après avoir recensé l'ensemble des besoins, nous distinguons les différents acteurs présentés dans le Tableau ?? suivant :

Acteur	Fonctions
Client	Les personnes et les entreprises qui ont besoin de transporter des marchandises d'un point A à un point B
Transporteur	Les propriétaires de véhicules et les transporteurs qui souhaitent proposer leurs services de transport de marchandises aux clients
Administrateur	L'administrateur est responsable de la gestion et de la supervision du système ou de l'application.

TABEAU 2.2 – Identification des acteurs du système.

2.4.2 Digramme de cas d'utilisation global

La Figure 2.1 suivante illustre le diagramme de cas d'utilisation global du système à réaliser :

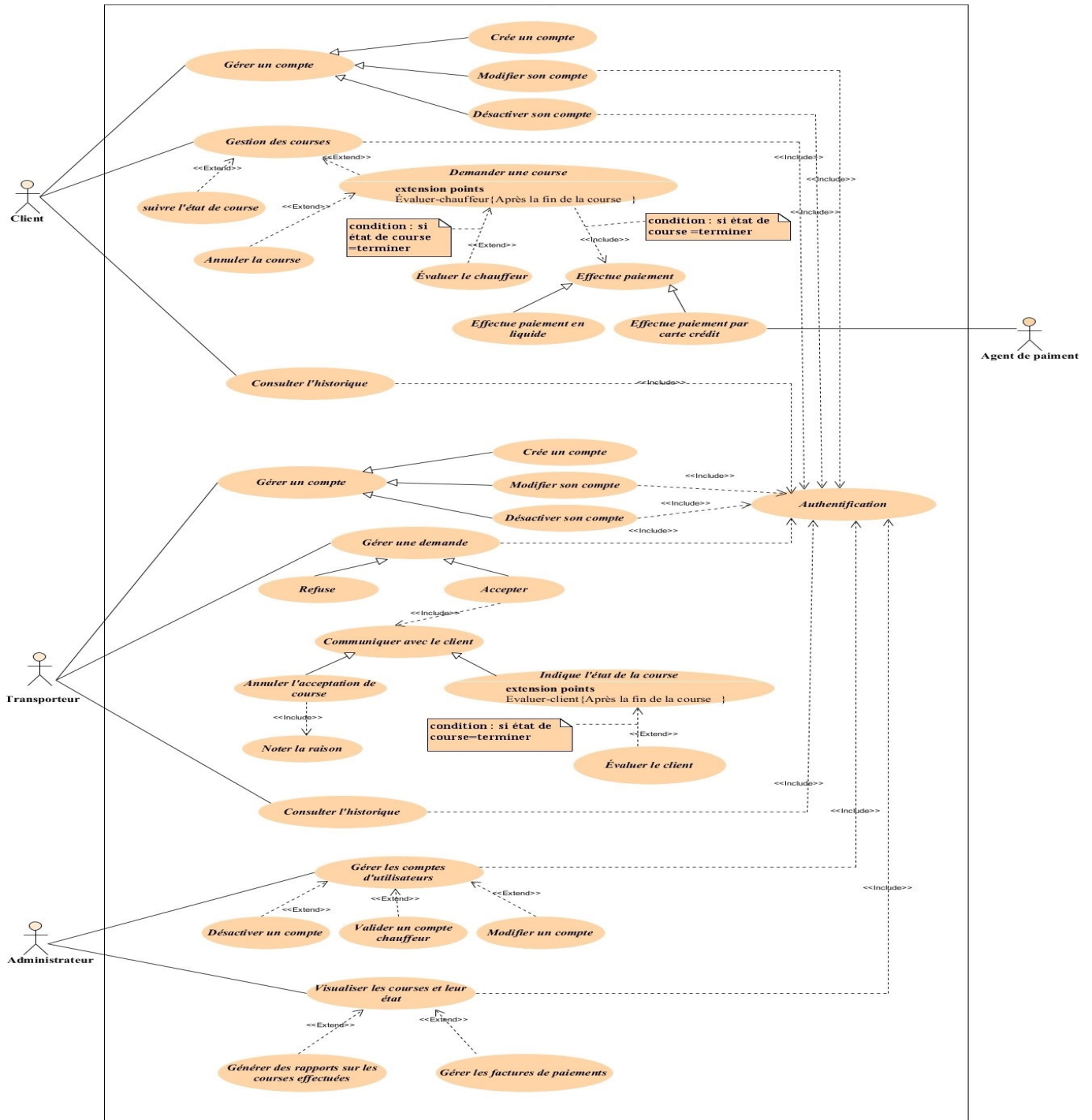


FIGURE 2.1 – Digramme de cas d'utilisation global.

2.5 Backlog du produit

Au démarrage du développement d'un produit agile, le projet doit être découpé en petites fonctionnalités ou tâches à réaliser pour faciliter la construction de l'application. De plus, cela permet à l'équipe de développement d'avoir une vision du produit à réaliser. Dans ce contexte, le Tableau suivant 2.3 présente les priorités et les estimations associées à chaque élément du Product Backlog

Product backlog				
Thème	ID U.S	Description User Story	Priorité	Estimation
Authentification et gestion des utilisateurs	1.1	En tant que client, je veux pouvoir créer un compte et me connecter pour accéder à mes demandes de transport.	1	4 jours
	1.2	En tant que transporteur, je veux pouvoir m'inscrire et me connecter pour accéder aux demandes de transport qui me sont assignées.	1	4 jours
	1.3	En tant qu'administrateur, je veux pouvoir gérer les comptes d'utilisateurs et les autorisations.	1	3 jours
Gestion des demandes de transport	2.1	En tant que client, je veux pouvoir soumettre une demande de transport en fournissant les détails de la marchandise, de la destination et des dates de livraison.	2	4 jours
	2.2	En tant que transporteur, je veux pouvoir accepter ou refuser une demande de transport.	2	3 jours
	2.3	En tant qu'administrateur, je veux pouvoir suivre l'état des demandes de transport et affecter des Transporteurs aux demandes acceptées.	2	3 jours
Gestion des livraisons et paiement	3.1	En tant que client, je veux suivre l'état de ma demande de transport	3	3 jours

	3.2	En tant que transporteur, je veux pouvoir mettre à jour l'état de la course	3	3 jours
	3.3	En tant qu'administrateur, je veux pouvoir générer des rapports sur les courses	3	2 jours
	3.4	En tant que client, je veux pouvoir effectuer des paiements en ligne ou espèces pour les frais de transport	3	4 jours

TABLEAU 2.3 – Présentation backlog du produit.

2.6 Planification des sprints

Le Tableau 2.4 suivant illustre la planification des sprints en ce qui concerne notre projet :

Sprint 1	Sprint 2	Sprint 3
Durée : 4 semaines	Durée : 4 semaines	Durée : 4 semaines
Authentification et gestion des utilisateurs	Gestion des demandes de transport	Gestion des livraisons et paiement

TABLEAU 2.4 – Planification des sprints.

2.7 Conclusion

En conclusion, ce chapitre a mis en place les bases essentielles de notre approche de développement Scrum pour le projet de création de notre startup. Nous avons présenté les acteurs clés, spécifié avec précision les besoins fonctionnels et non fonctionnels, ainsi que modélisé ces besoins à travers un diagramme de cas d'utilisation. De plus, la création du Product Backlog et

la planification des sprints ont été abordées, marquant ainsi le démarrage concret de notre projet. Le chapitre suivant approfondira notre exploration de la méthodologie Scrum en détaillant la mise en œuvre pratique de ces concepts.

CHAPITRE 3

CONCEPTION

3.1 Introduction

Le chapitre de conception marque une étape cruciale dans notre projet, où nous traduisons les besoins identifiés précédemment en une structure solide pour notre application. Dans cette phase, nous examinons les résultats de nos trois sprints, chacun apportant sa propre contribution à la vision globale. Nous avons consolidé ces informations pour créer un plan détaillé qui guidera la création de notre application. Dans ce chapitre, nous allons explorer les principaux éléments de notre conception, y compris les diagrammes de cas d'utilisation, les diagrammes de classes, les modèles relationnels et les maquettes. Notre objectif est de garantir que notre application répondra aux besoins de nos utilisateurs de manière efficace et intuitive, tout en offrant une architecture robuste pour le développement à venir.

3.2 Étude du premier sprint

La durée prévue pour ce premier sprint est de 22 jours, et il est composé des items suivant

- Inscription.
- Authentification.
- Gérer les comptes des utilisateurs par l'administrateur (valider, modifier, désactiver).

3.2.1 User story

Nous présentons dans la Figure 3.1, les fonctionnalités à développer qui définissent des user stories. Ces dernières, est une description courte et simple d'une fonctionnalité ou d'un besoin du point de vue de l'utilisateur

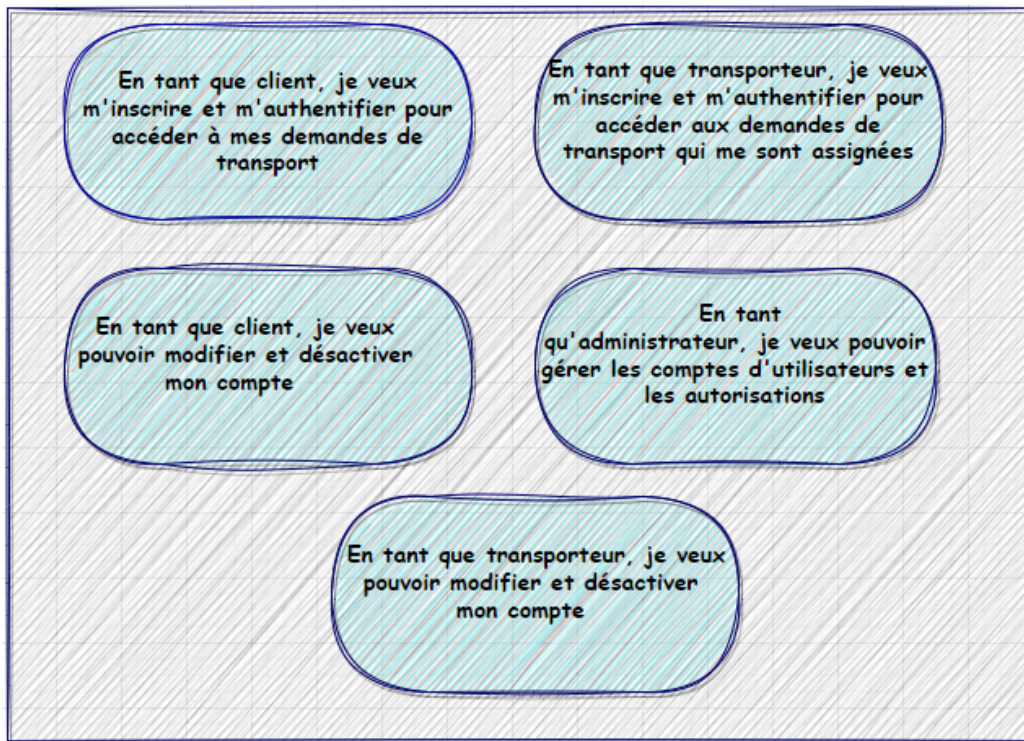


FIGURE 3.1 – User Story du sprint 1.

3.2.2 Décomposition du sprint 1

Pendant toute cette période, nous avons travaillé sur les User Stories de ce sprint afin de créer un incrément potentiellement livrable. La Figure 3.2, illustre la répartition du temps pour le Sprint 1.

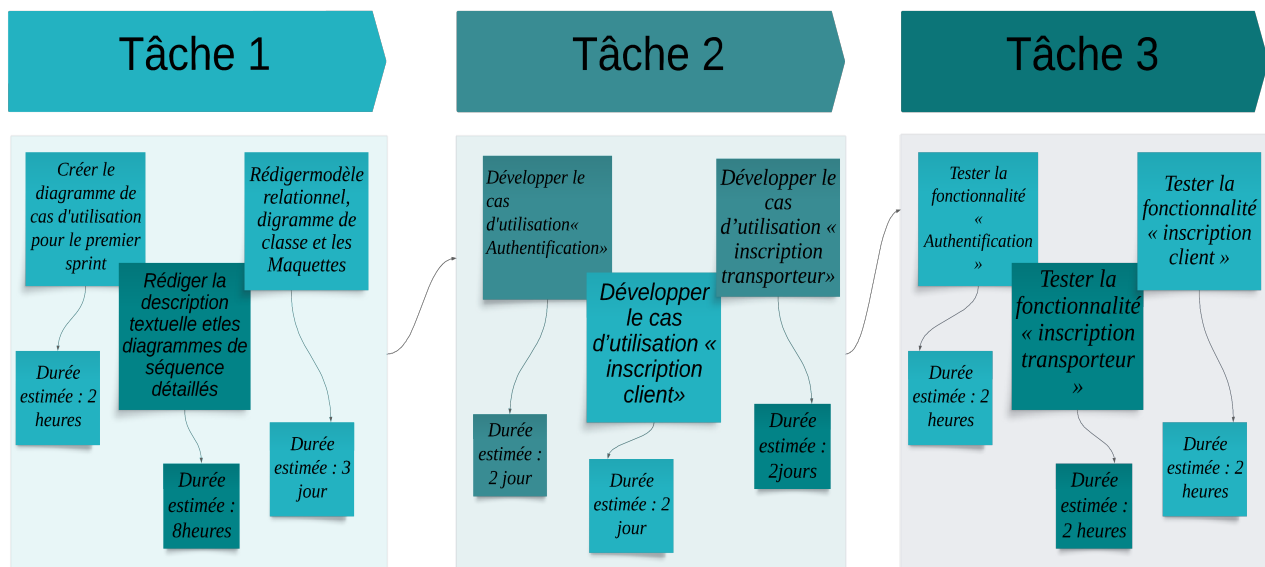


FIGURE 3.2 – Décomposition du sprint 1 en termes de temps.

3.2.3 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation de la Figure 3.3 décrit d'une façon globale les cas d'utilisations composant le sprint 1.

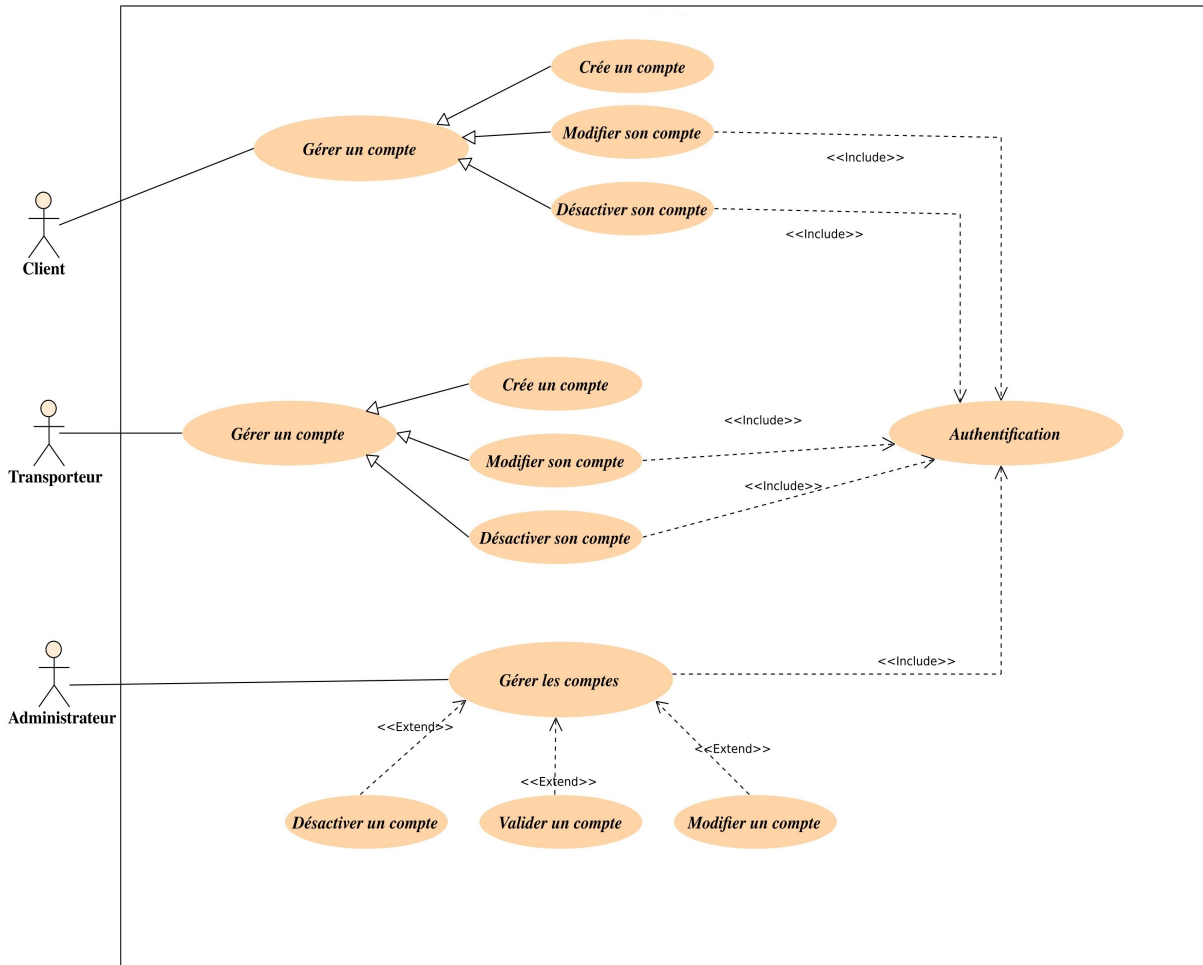


FIGURE 3.3 – Diagramme de cas d'utilisation du sprint 1.

3.2.4 Cas d'utilisation « Inscription client »

Description textuelle

Chaque cas d'utilisation d'un système doit être défini textuellement. Nous avons choisi de décrire ici le cas d'utilisation « Inscription client ».

Le Tableau 3.1 suivant illustre la description textuelle du cas d'utilisation « Inscription client »

Sommaire d'inscription	
Titre	Inscription d'un client
Résumé	En tant que client, je peux m'inscrire dans l'application
Acteurs	Client : personne désirant utiliser les services de transport proposés.
Description des scénarios	
Pré-conditions	L'utilisateur accède à la page d'inscription.
Scénario nominal	Le client accède à la page d'inscription de la plateforme.
	Le client remplit les champs obligatoires du formulaire d'inscription, tels que son nom, prénom, numéro de téléphone, adresse e-mail et mot de passe [A1].
	Le client soumet le formulaire d'inscription.
	Le système vérifie que les informations fournies sont valides et non déjà utilisées par un autre utilisateur.
	En cas de succès, le système enregistre les informations du client et crée son compte.
	Le client reçoit une notification ou un e-mail de confirmation d'inscription avec les détails de son compte.
Enchaînement alternative	[A1] Si les informations fournies par le client sont invalides ou déjà utilisées, le système affiche un message d'erreur et demande au client de corriger les champs concernés.
Post-conditions	Le client est maintenant inscrit sur la plateforme et peut se connecter à son compte pour accéder aux fonctionnalités réservées aux utilisateurs enregistrés.

TABLEAU 3.1 – Description textuelle du cas d'utilisation «Inscription client»

Diagramme de séquence détaillé

La Figure 3.4 représente le diagramme de séquence détaillé du cas d'utilisation « Inscription client »

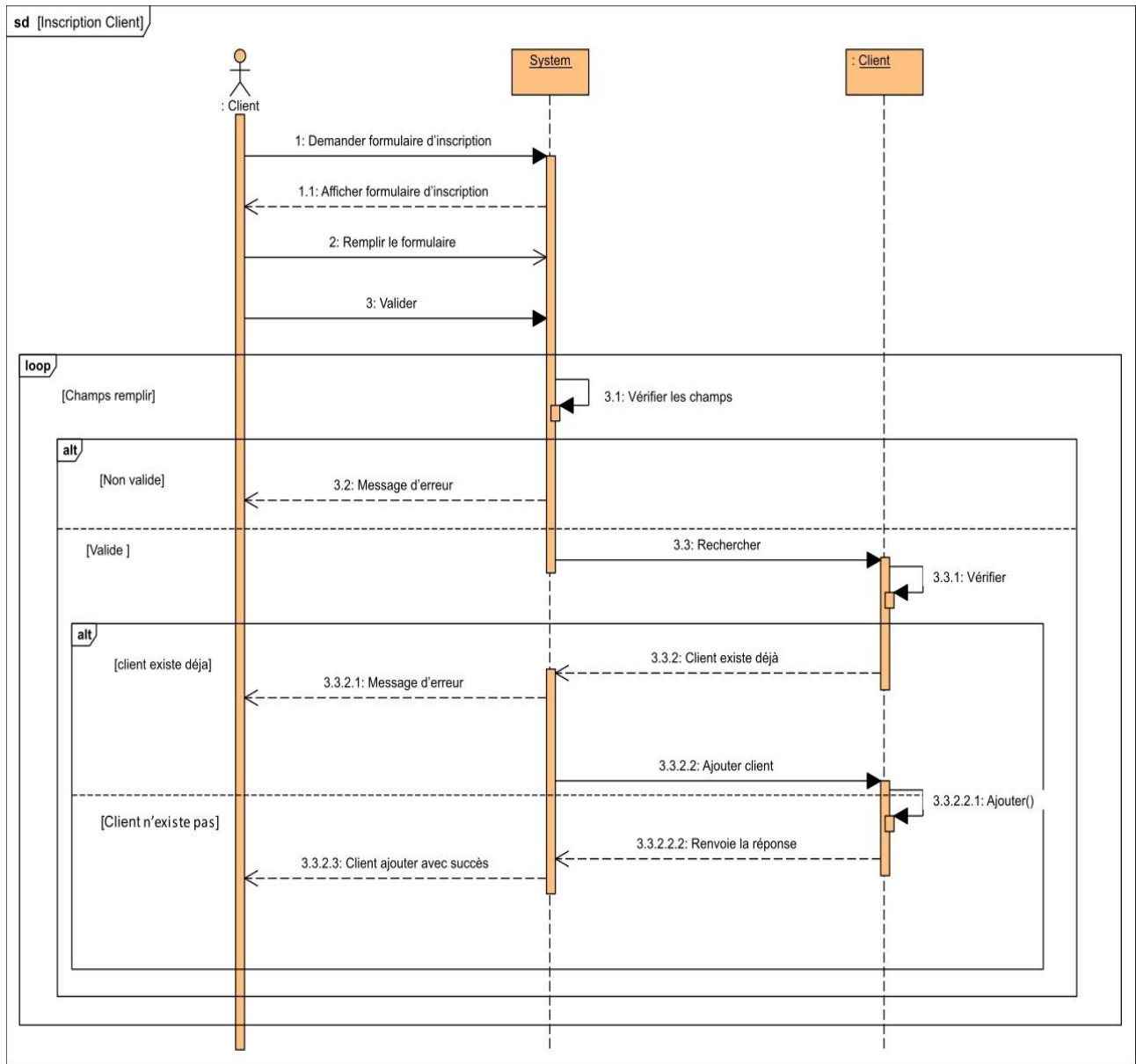


FIGURE 3.4 – Diagramme de séquence détaillé du cas «Inscription client»

3.2.5 Cas d'utilisation « Inscription transporteur»

Description textuelle

Le Tableau 3.2 suivant illustre la description textuelle du cas d'utilisation «Inscription transporteur».

Sommaire d'inscription	
Titre	Inscription du transporteur
Résumé	En tant que transporteur, je peux m'inscrire dans l'application
Acteurs	Transporteur
Description des scénarios	
Pré-conditions	L'utilisateur accède à la page d'inscription.
Scénario nominal	Le transporteur accède à l'application ou au système d'inscription des transporteurs.
	Le système affiche le formulaire d'inscription du transporteur.
	Le transporteur remplit tous les champs obligatoires du formulaire avec les informations requises [A1] [A2] [A3].
	Le transporteur valide le formulaire d'inscription.
	Le système vérifie que tous les champs obligatoires sont remplis correctement.
	L'administrateur est notifié de l'inscription du nouveau transporteur et doit valider son compte avant que le transporteur puisse pleinement utiliser l'application.
	Le transporteur doit attendre la validation de son compte par l'administrateur avant de pouvoir accéder à toutes les fonctionnalités de l'application
Enchaînement alternative	[A1] Si des champs obligatoires ne sont pas remplis, le système affiche un message d'erreur demandant au transporteur de les remplir.
	[A2] Si les informations fournies par le transporteur ne sont pas valides, le système affiche un message d'erreur demandant au transporteur de les corriger.
	[A3] Si les informations fournies par le Transporteur correspondent à un compte déjà existant, le système affiche un message d'erreur demandant au transporteur de modifier les informations ou de récupérer son compte existant.
Post-conditions	Le transporteur est enregistré avec succès dans le système.
	Les informations du transporteur et de son véhicule sont enregistrées dans la base de données.

<p>Le compte du transporteur est activé et prêt à être utilisé.</p> <p>Le transporteur peut accéder aux fonctionnalités réservées aux Transporteurs , telles que la gestion des demandes de transport.</p>
--

TABLEAU 3.2 – Description textuelle du cas d'utilisation "Inscription transporteur"

Diagramme de séquence détaillé

La Figure3.5 représente le diagramme de séquence détaillé du cas d'utilisation « Inscription transporteur ».

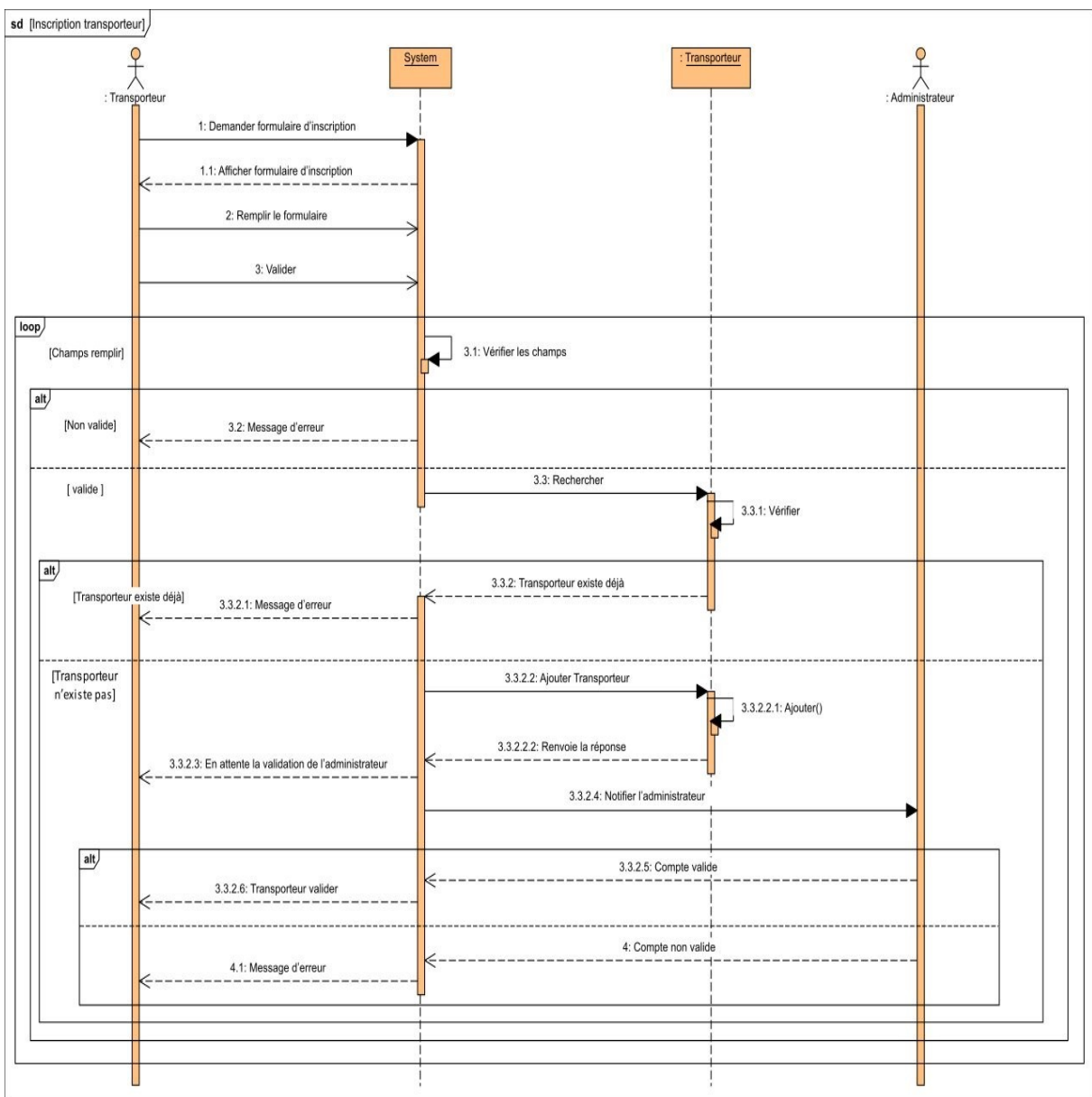


FIGURE 3.5 – Diagramme de séquence détaillé du cas «Inscription transporteur».

3.2.6 Cas d'utilisation « S'authentifier »

Description textuelle

Le Tableau 3.3 suivant illustre la description textuelle du cas d'utilisation «S'authentifier».

Sommaire d'identification	
Titre	Authentification
Résumé	L'authentification permet d'identifier les utilisateurs du système.
Acteurs	Client, Transporteur
Description des scénarios	
Pré-conditions	L'utilisateur doit déjà être inscrit.
Scénario nominal	L'utilisateur accède à l'interface de connexion de l'application.
	L'application affiche les champs de saisie pour adresse e-mail (ou numéro téléphone) et le mot de passe.
	L'utilisateur saisit son adresse e-mail (ou numéro téléphone) et son mot de passe [A1].
	L'application vérifie les informations d'identification fournies par l'utilisateur.
	Si les informations sont valides, l'application considère l'utilisateur comme authentifié et lui donne accès aux fonctionnalités de l'application.
	L'application affiche l'interface principale de l'application avec les fonctionnalités disponibles pour l'utilisateur authentifié.
Enchaînement alternative	[A1] Si les informations d'identification sont incorrectes, l'application affiche un message d'erreur indiquant que les informations d'identification sont invalides.
Post-conditions	L'utilisateur est authentifié et peut accéder aux fonctionnalités de l'application.
	Si une erreur d'authentification se produit, l'utilisateur reste sur l'interface de connexion et reçoit un message d'erreur approprié.

TABLEAU 3.3 – Description textuelle du cas d'utilisation "Authentification"

Diagramme de séquence détaillé

La Figure 3.6 représente le diagramme de séquence détaillé du cas d'utilisation « S’authentifier ».

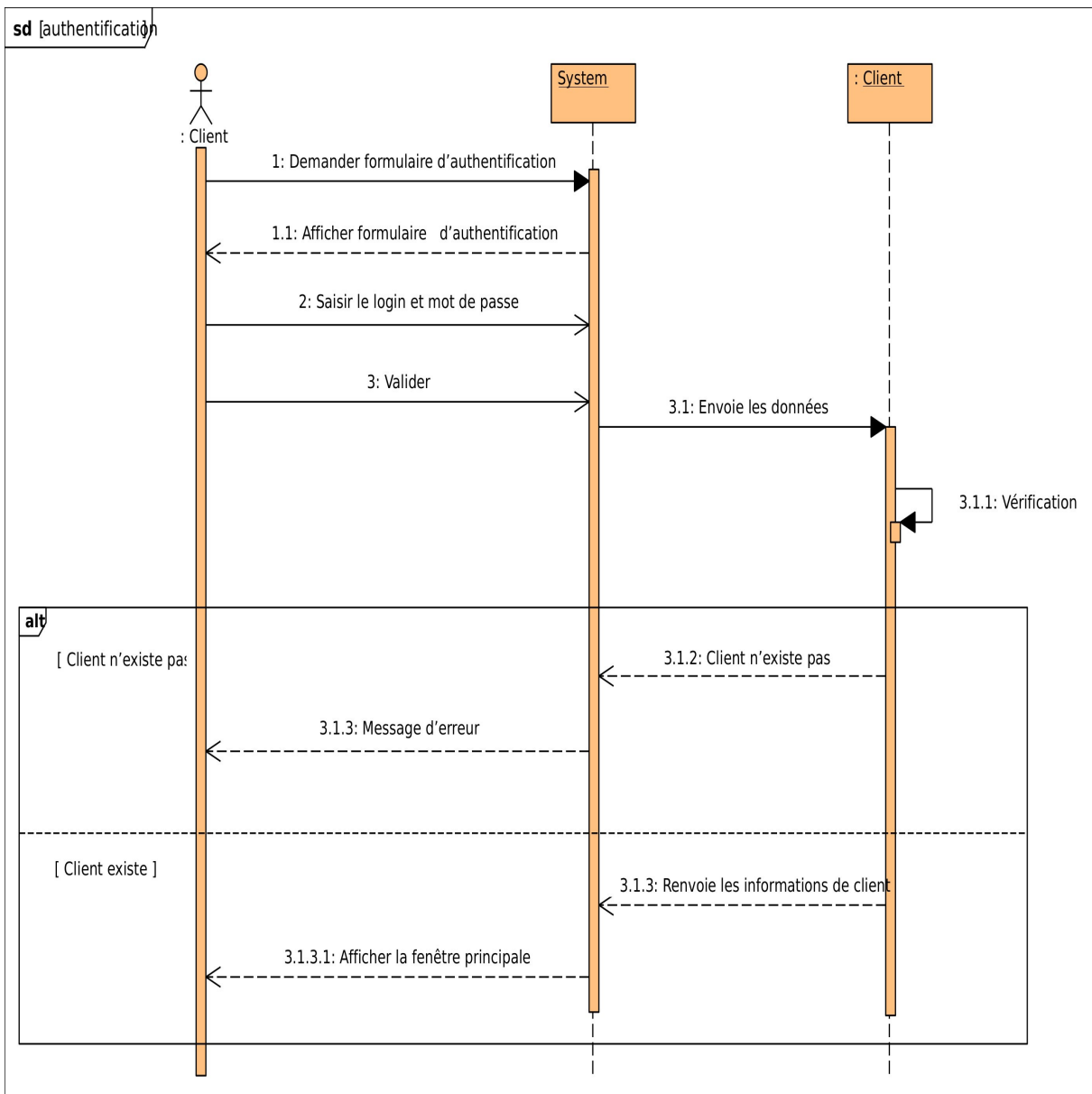


FIGURE 3.6 – Diagramme de séquence détaillé du cas « S’authentifier ».

3.2.7 Diagramme de classes

La Figure 3.7 suivante représente le diagramme de classes issu du Sprint 1.

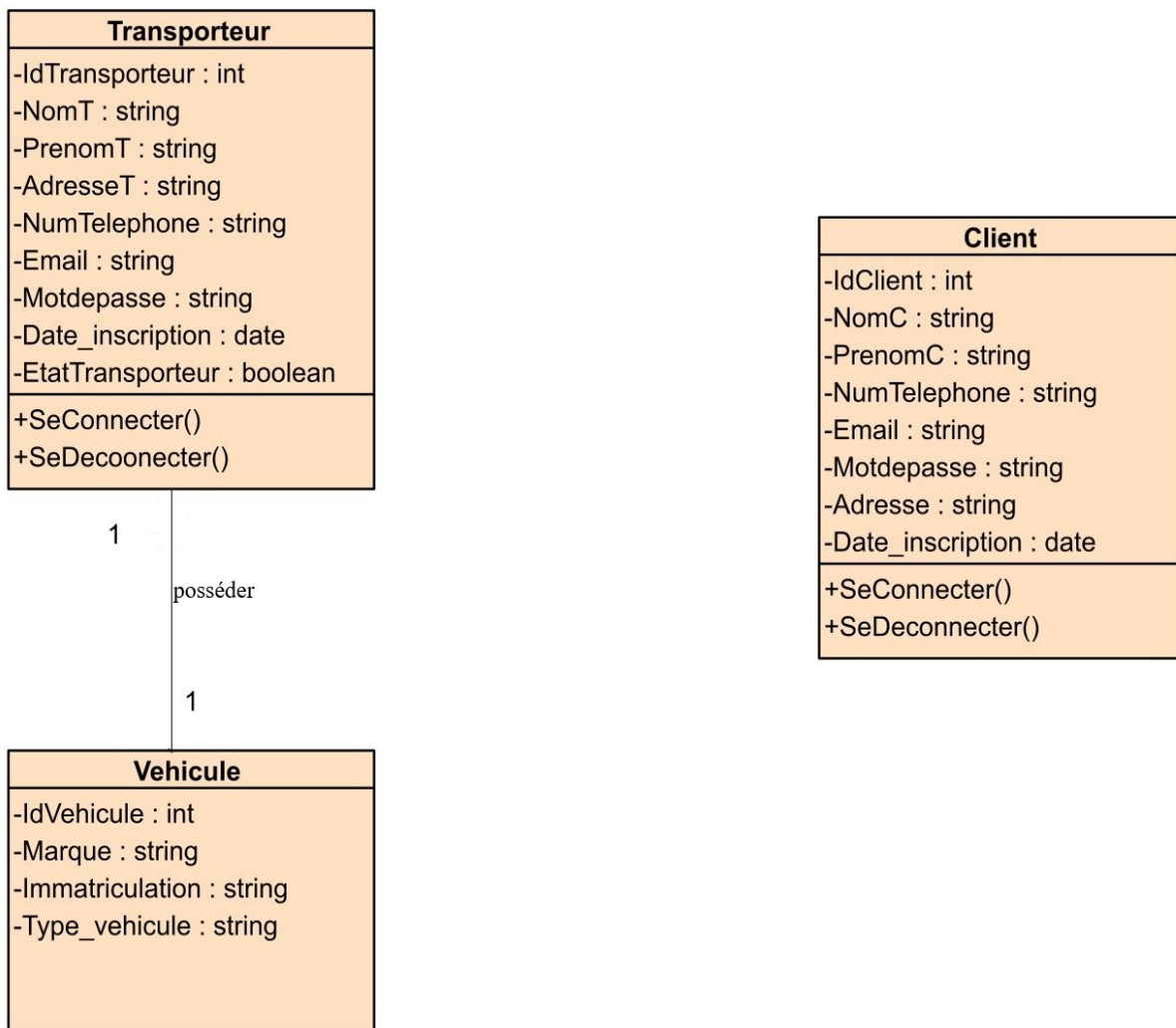


FIGURE 3.7 – Diagramme de class du sprint 1.

3.2.8 Modèle relationnel

Voici le modèle relationnel correspondant au diagramme de classe du sprint 1

Client (idClient, nomC, prenomC, numTelephone, email, motdepasse, adresse, date_Inscription)

Transporteur(idTransporteur, nomT, prenomT, adresseT, numTelephone, email, motdepasse, date_Inscription, EtatTransporteur)

Vehicule (idVehicule, #IdTransporteur, marque, immatriculation, type_vehicule,)

3.2.9 Maquette

Interfaces d'Inscription

Les maquettes d'inscription sont soigneusement élaborées pour deux catégories d'utilisateurs essentielles : les clients et les transporteurs. Pour les clients, l'accent est mis sur la simplicité et la convivialité, offrant un processus d'inscription rapide pour accéder aux services. Les transporteurs, quant à eux, disposent d'une maquette d'inscription adaptée à leurs besoins spécifiques, incluant des informations relatives au véhicule, aux documents légaux et aux préférences de livraison. Ces maquettes d'inscription visent à garantir une expérience fluide et convaincante pour chaque groupe d'utilisateurs, facilitant ainsi leur intégration et leur utilisation de la plateforme.

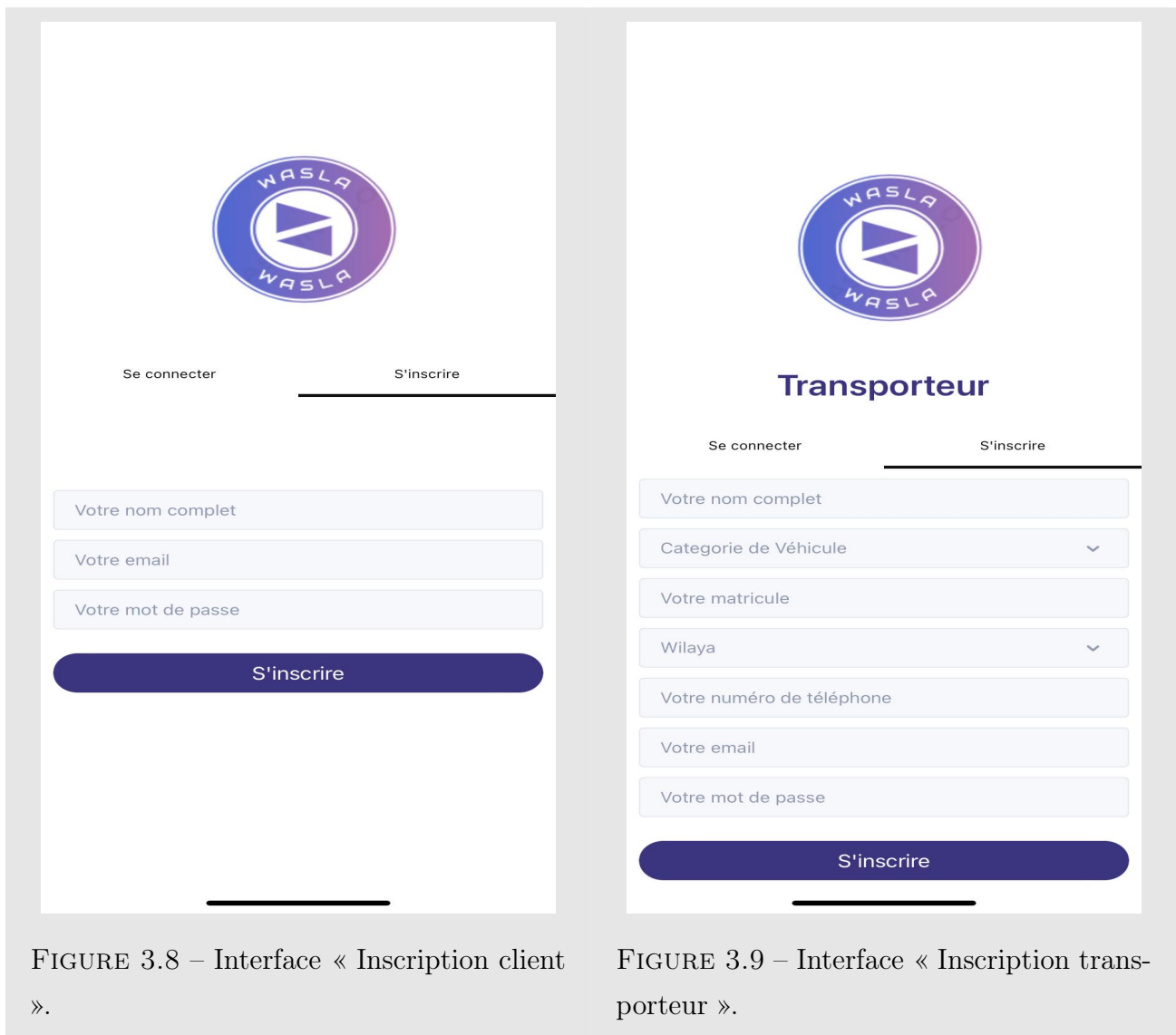


FIGURE 3.8 – Interface « Inscription client ».

FIGURE 3.9 – Interface « Inscription transporteur ».

Interfaces d'Authentification

Notre application offre une variété de maquettes d'authentification adaptées aux besoins de chaque type d'utilisateur. Pour nos clients, nous privilégions la simplicité et la convivialité, leur permettant de se connecter rapidement pour accéder à nos services. Les transporteurs bénéficient d'une maquette qui prend en compte les exigences de leur activité, tandis que nos administrateurs ont accès à des mesures de sécurité renforcées pour gérer efficacement la plateforme. Chacune de ces interfaces a été soigneusement élaborée pour garantir un accès sécurisé et une expérience utilisateur optimale

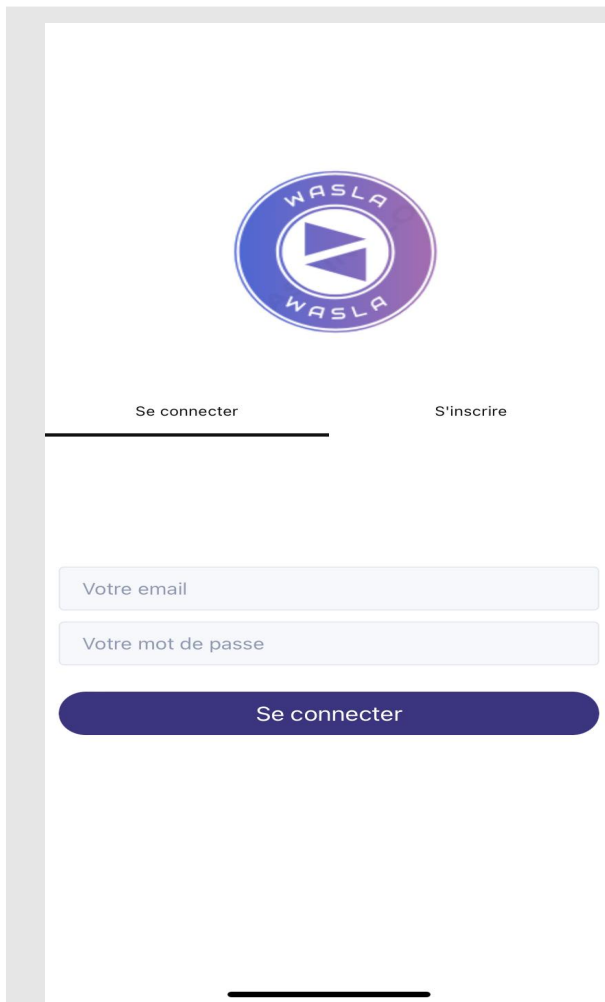


FIGURE 3.10 – Interface « Authentification client ».

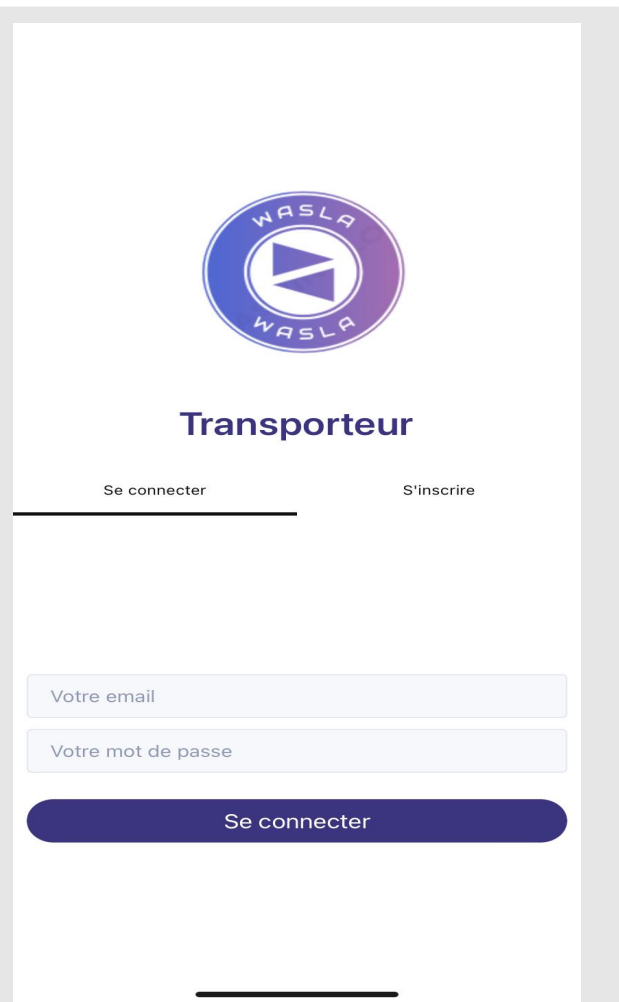


FIGURE 3.11 – Interface « Authentification transporteur ».

L'interface d'administrateur

L'interface d'administration permet à l'administrateur d'accéder à une liste de transporteur où il peut valider ou suspendre chaque transporteur en un simple clic, offrant ainsi une gestion efficace des chauffeurs de l'entreprise.

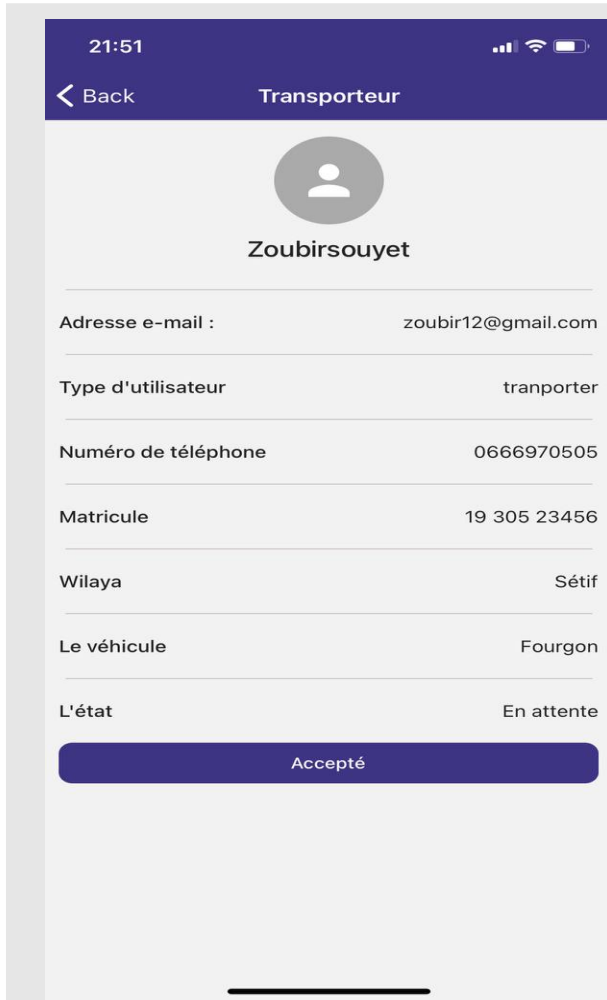


FIGURE 3.12 – Interface «valider transporteur ».



FIGURE 3.13 – Interface « suspendre transporteur ».

3.3 Étude du deuxième sprint

La durée prévue pour ce deuxième sprint est de 25 jours, et il est composé des items suivant

- Demander une course .
- Annuler la course .
- Evaluer le Transporteur .
- Visualiser les demandes de course .

3.3.1 User story

Nous présentons dans la Figure 3.14, les fonctionnalités à développer qui définissent des user stories. Ces dernières, est une description courte et simple d'une fonctionnalité ou d'un besoin du point de vue de l'utilisateur

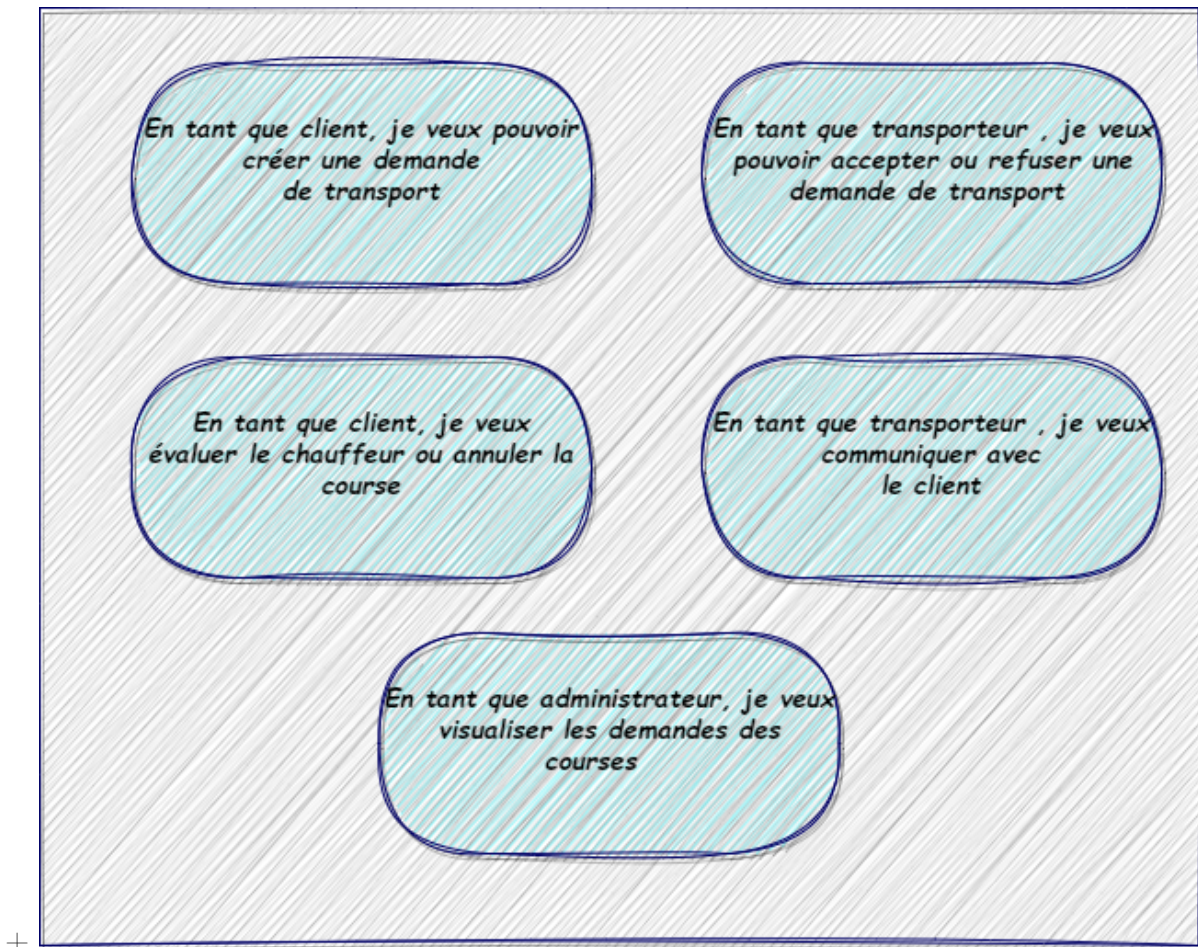


FIGURE 3.14 – User Story du sprint 2.

3.3.2 Décomposition du sprint 2

Pendant toute cette période, nous avons travaillé sur les User Stories de ce sprint afin de créer un incrément potentiellement livrable. La Figure 3.15 illustre la répartition du temps pour le Sprint 2.

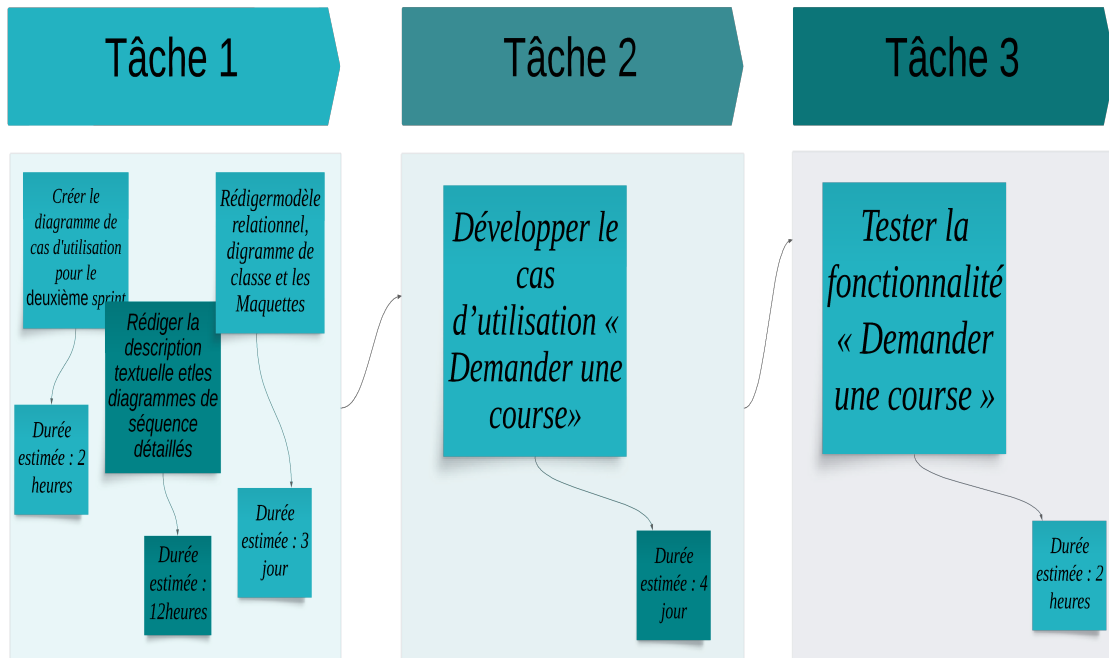


FIGURE 3.15 – Décomposition du sprint 2 en termes de temps.

3.3.3 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation de la Figure 3.16 décrit d'une façon globale les cas d'utilisations composant du sprint 2.

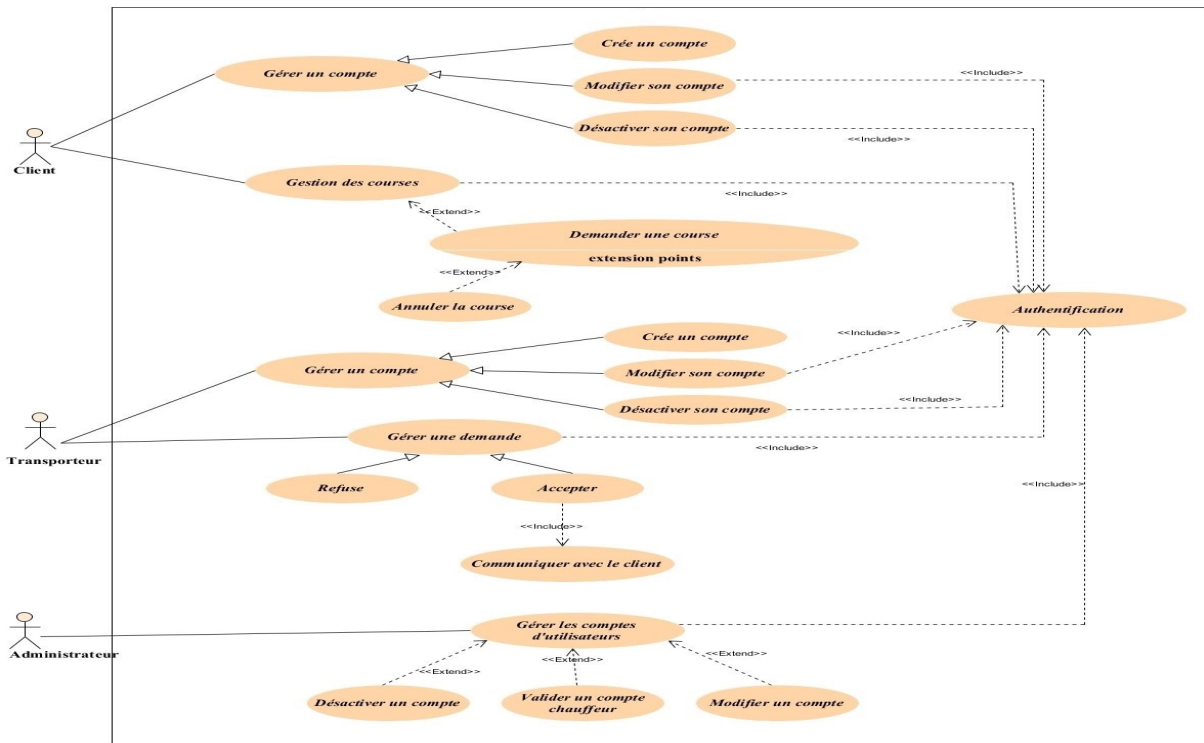


FIGURE 3.16 – Diagramme de cas d'utilisation du sprint 2.

3.3.4 Cas d'utilisation «Demander une course»

Description textuelle

Le Tableau 3.4 suivant illustre la description textuelle du cas d'utilisation «Demander une course».

Sommaire de demande course	
Titre	Demander une course
Résumé	Client peut faire une demande de course
Acteurs	Client
Description des scénarios	
Préconditions	Le client s'est authentifié avec succès sur l'application.
Scénario nominal	Le client accède à son compte sur l'application après s'être authentifié.
	Le client clique sur le bouton "Demander une course".
	Le système affiche une carte géographique avec deux champs d'entrée : "Départ" et "Destination".

	Le client remplit les champs en spécifiant le lieu de départ et le lieu de destination [A1].
	Le système vérifie que tous les champs sont remplis et valides.
	Si les champs sont vérifiés et validés par le client, le système affiche une deuxième interface avec une description de la marchandise à transporter.
	Le client remplit les informations concernant la marchandise [A2].
	Le système vérifie que tous les champs de la description de la marchandise sont remplis et valides.
	Si tous les champs sont remplis et valides, le système recherche les transporteurs disponibles.
	Le système notifie les transporteurs disponibles de la demande de course.
	Un transporteur disponible reçoit une notification de la demande et peut accepter ou refuser.
	Si le transporteur accepte, le système permet au transporteur de communiquer avec le client pour discuter des détails de la course.
	Le système notifie le client qu'un transporteur a été trouvé et qu'il a accepté la demande.
	Le client peut communiquer avec le transporteur pour finaliser les détails de la course.
Enchaînement alternative	[A1] Si le client ne remplit pas tous les champs requis sur la carte géographique, le système affiche un message d'erreur demandant de remplir tous les champs obligatoires.
	[A2] Si les champs de la description de la marchandise ne sont pas tous remplis, le système affiche un message d'erreur demandant de fournir toutes les informations nécessaires.
Post-conditions	Le client a fait une demande de course avec succès.

TABLEAU 3.4 – Description textuelle du cas d'utilisation "Demander une course"

Diagramme de séquence détaillé

La Figure 3.17 représente le diagramme de séquence détaillé du cas d'utilisation «Demander une course».

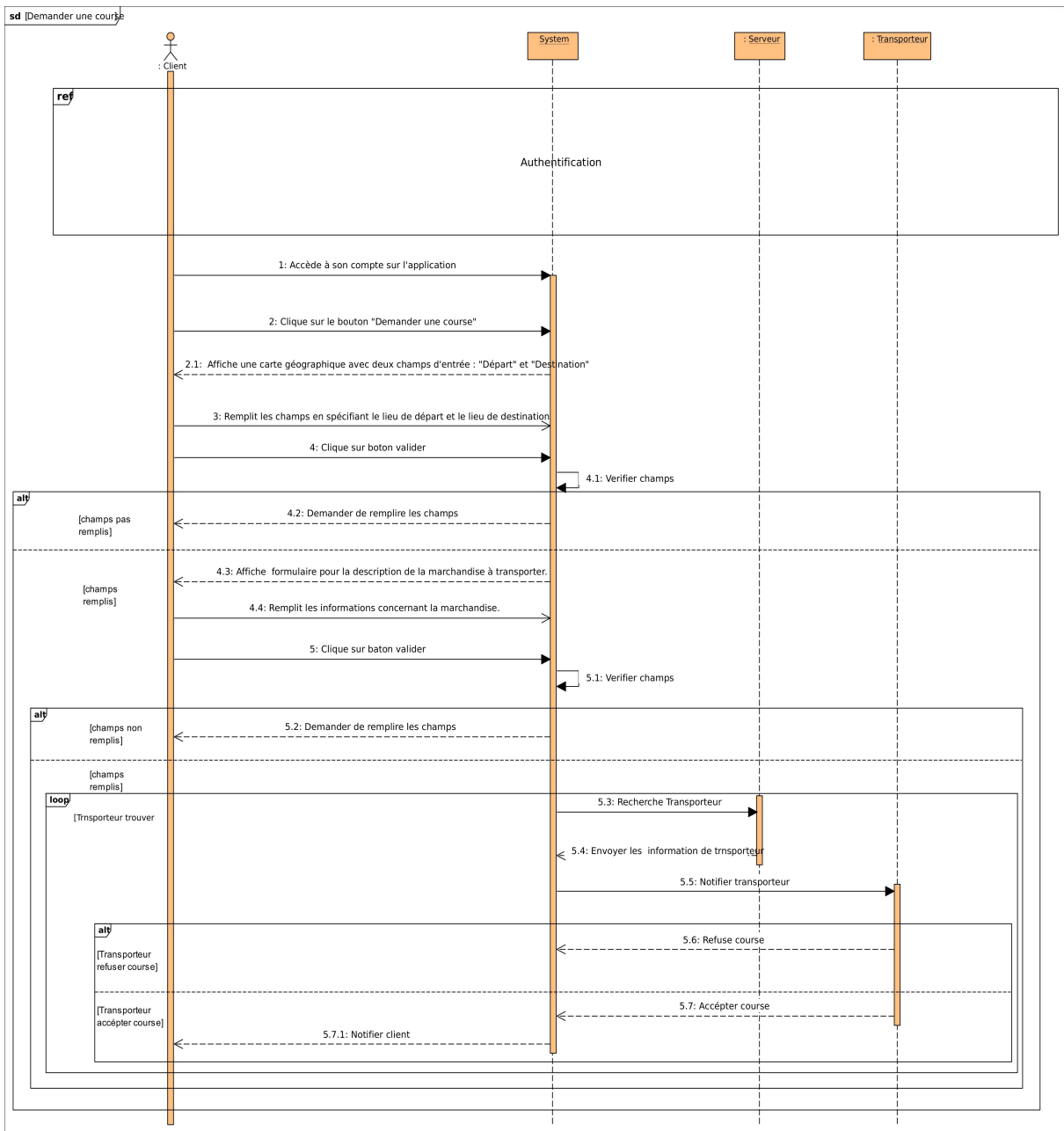


FIGURE 3.17 – Diagramme de séquence détaillé du cas «Demander une course».

3.3.5 Diagramme de classe

La Figure 3.18 suivante représente le diagramme de classes issu du Sprint 2.

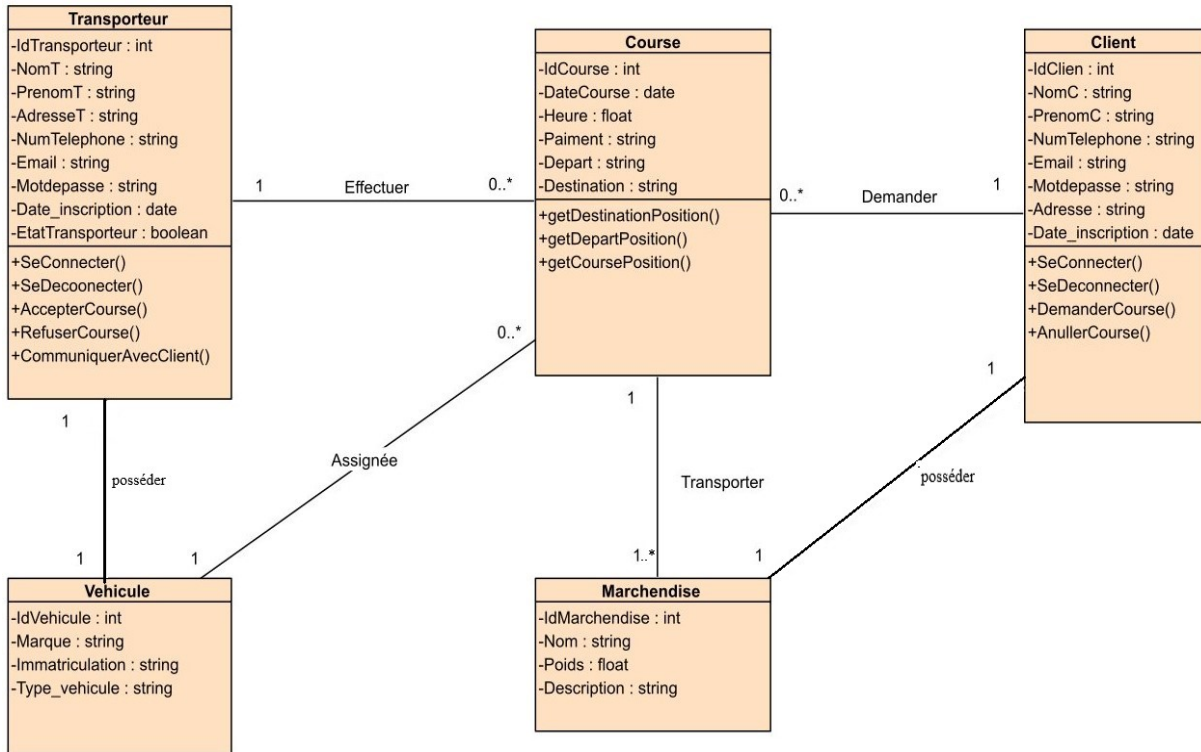


FIGURE 3.18 – Diagramme de classe du sprint 2.

3.3.6 Modèle relationnel

Voici le modèle relationnel correspondant au diagramme de classe du sprint 2

Transporteur(idTransporteur, nomT, prenomT, adresseT, numTelephone, email, motdepasse, date_inscription, etatTransporteur).

Vehicule(idVehicule, #idTransporteur, marque, immatriculation, type_vehicule).

Client(idClient, nomC, prenomC, numTelephone, email, maotdepasse, adresse, date_inscription).

marchandise(idMarchandise,#idClient, nom, poids, description).

Course(idCourse, dateCourse, heure, paiement, depart, destination, #idVehicule, #idMarchandise, #idClient,#idTransporteur).

3.3.7 Maquette

Interface de demande de Course

Notre interface simplifiée permet aux utilisateurs de demander un transport en fournissant les détails essentiels de leur course, tels que le point de départ et d'arrivée.

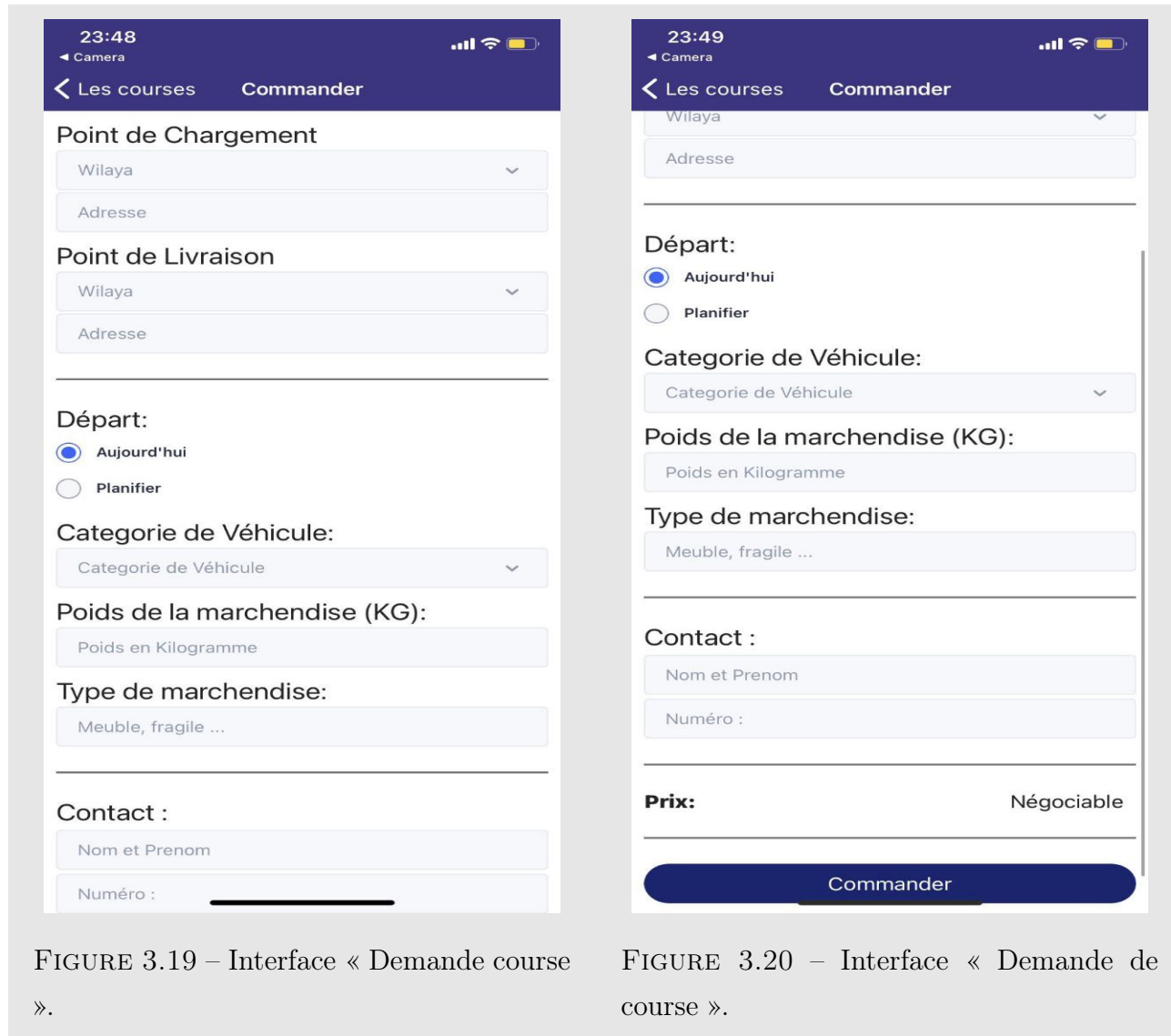


FIGURE 3.19 – Interface « Demande course ».

FIGURE 3.20 – Interface « Demande de course ».

3.3.8 Gestion des Courses et Acceptation

L'interface du transporteur offre une liste des courses disponibles, affichant les détails essentiels tel que l'emplacement de prise en charge, la destination. Le transporteur peut ainsi parcourir ces offres, choisir une course qui l'intéresse, puis appuyer sur un bouton "Accepter" pour confirmer son intention de prendre en charge cette course spécifique.

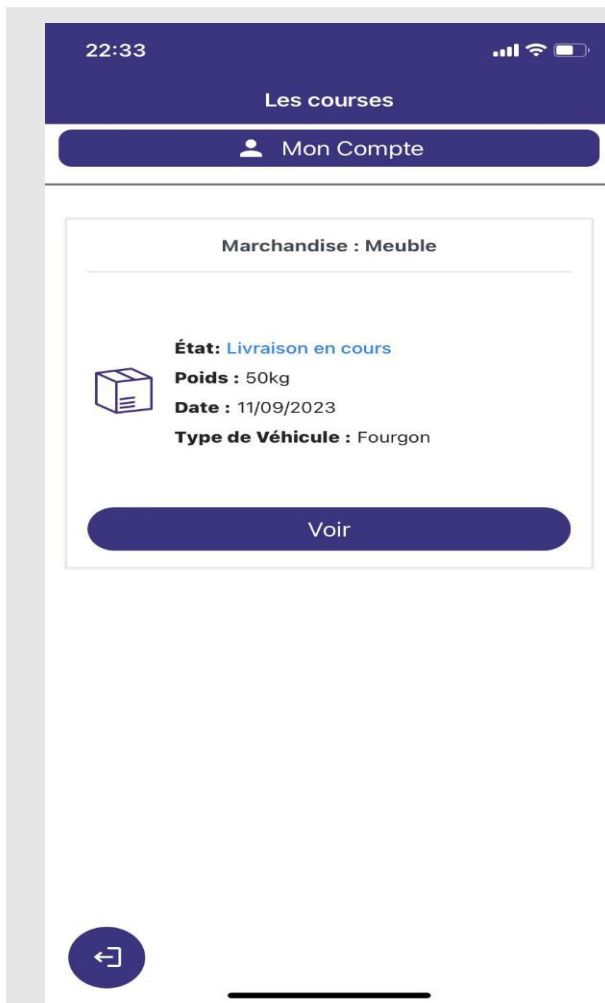


FIGURE 3.21 – Interface « Liste des course coté transporteur ».



FIGURE 3.22 – Dnterface « Detaille de course ».

3.4 Étude du Troisième sprint

La durée prévue pour ce deuxième sprint est de 25 jours, et il est composé des items suivant

- Suivre l'état de course .
- Effectue paiement .
- Annuler l'acceptation de course .
- Indiquer l'état de course .
- Visualiser les courses et leur état .

3.4.1 User story

Nous présentons dans la Figure 3.23, les fonctionnalités à développer qui définissent des user stories. Ces dernières, est une description courte et simple d'une fonctionnalité ou d'un besoin

du point de vue de l'utilisateur

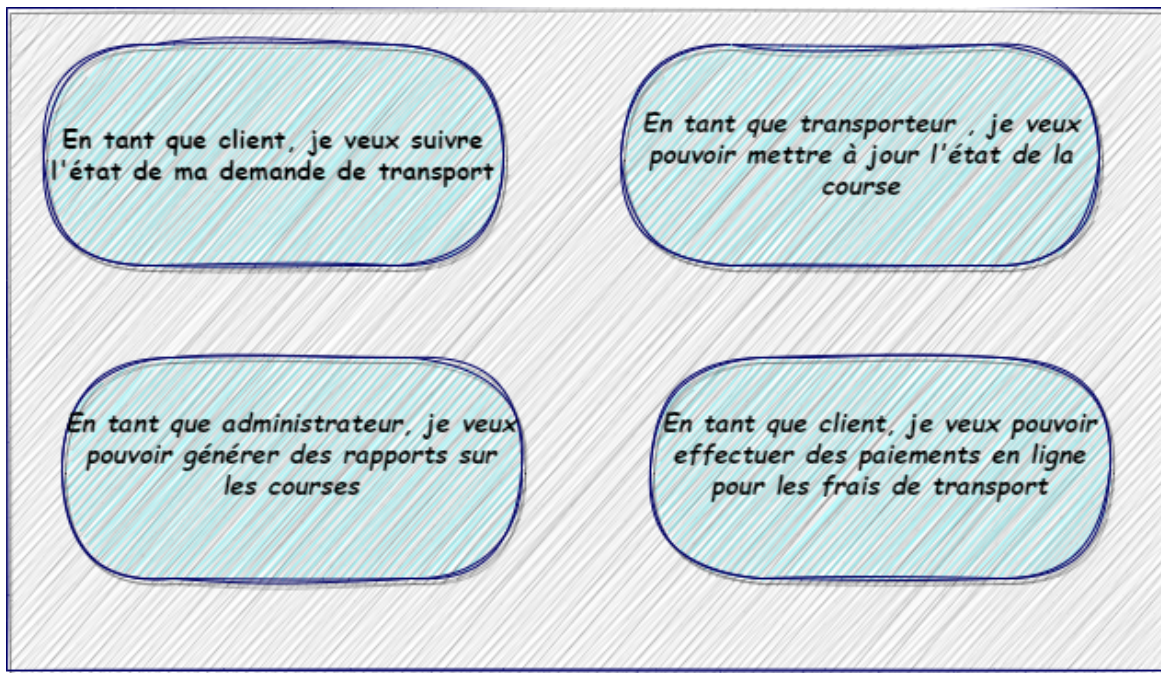


FIGURE 3.23 – User Story du sprint 3.

3.4.2 Décomposition du sprint 3

Pendant toute cette période, nous avons travaillé sur les User Stories de ce sprint afin de créer un incrément potentiellement livrable. La Figure 3.24 illustre la répartition du temps pour le Sprint 3.



FIGURE 3.24 – User Story du sprint 3.

3.4.3 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation de la figure 3.25 décrit d'une façon globale les cas d'utilisations composant du sprint 3.

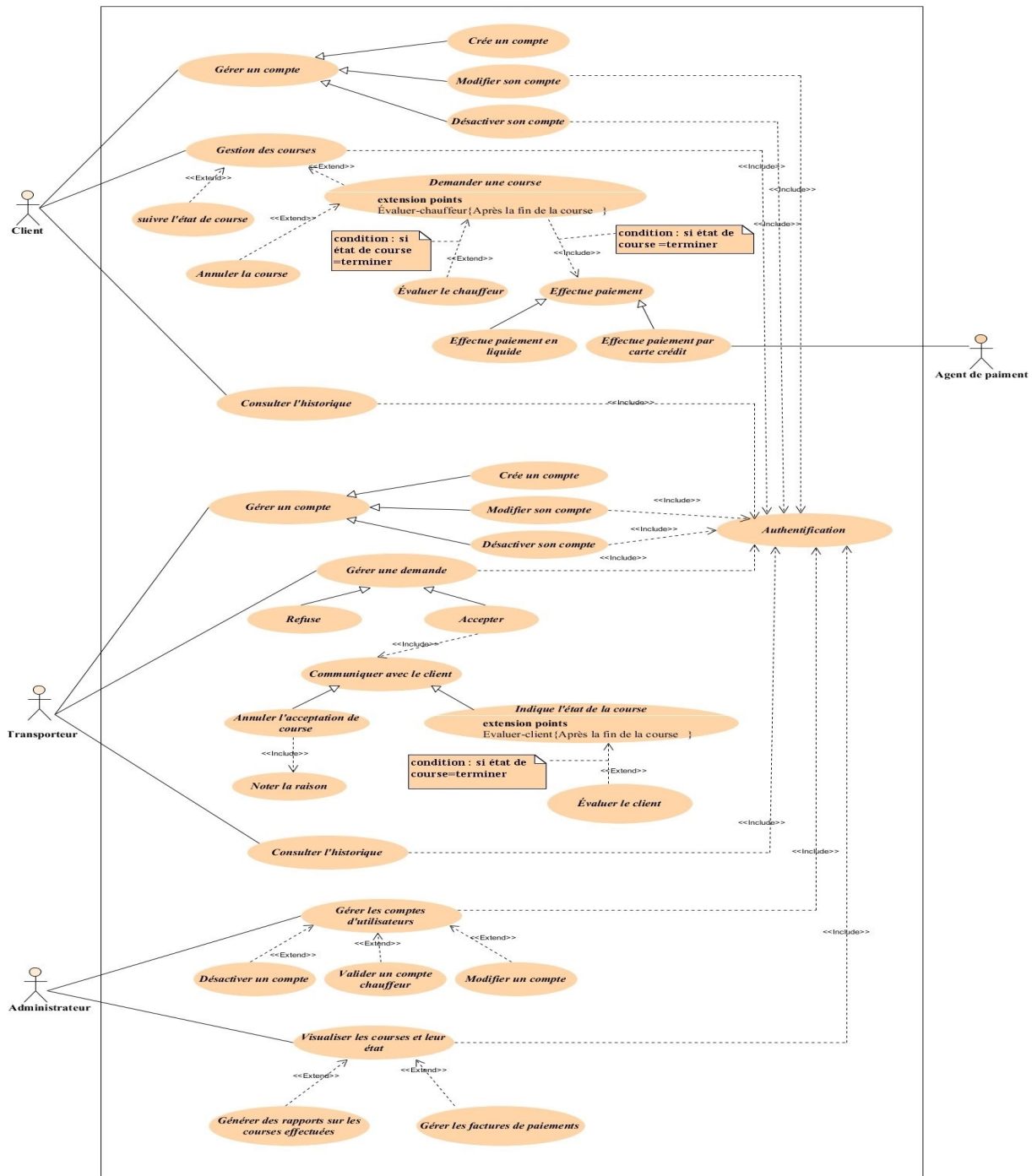


FIGURE 3.25 – Diagramme de cas d'utilisation du sprint 3.

3.4.4 Cas d'utilisation «Suivre état de course»

Description textuelle

Le Tableau 3.5 suivant illustre la description textuelle du cas d'utilisation «suivre état de course»

Sommaire suivre état de course	
Titre	Suivi en temps réel d'une course de transport de marchandise
Résumé	En tant que client, je veux suivre mn marchandise en temps réel
Acteurs	Client
Description des scénarios	
Préconditions	Le client s'est connecté à l'application ou au site web du système.
	Le client a sélectionné une demande de transport spécifique pour laquelle il souhaite suivre la marchandise en temps réel.
Scénario nominal	Le client accède à son compte sur l'application.
	Le système vérifie les informations d'identification du client dans la base de données.
	Le client sélectionne la demande de transport spécifique pour laquelle il souhaite suivre la marchandise en temps réel.
	Le système interroge la base de données pour récupérer les détails de la demande de transport, tels que l'identifiant de la course, les coordonnées de départ et d'arrivée, et les informations du Transporteur .
	Le système affiche les détails de la demande de transport au client.
Enchaînement d'erreur	Le système rencontre un problème lors de la récupération des détails de la demande de transport depuis la base de données.
	Le système affiche un message d'erreur indiquant qu'il est impossible de récupérer les détails de la demande de transport.
Postconditions	Le client est maintenant inscrit sur la plateforme et peut se connecter à son compte pour accéder aux
	Le client a sélectionné une demande de transport spécifique pour laquelle il souhaite suivre la marchandise en temps réel.

TABLEAU 3.5 – Description textuelle du cas d'utilisation "Suivre état de course"

Diagramme de séquence détaillé

La Figure 3.26 représente le diagramme de séquence détaillé du cas d'utilisation «Suivre état de course».

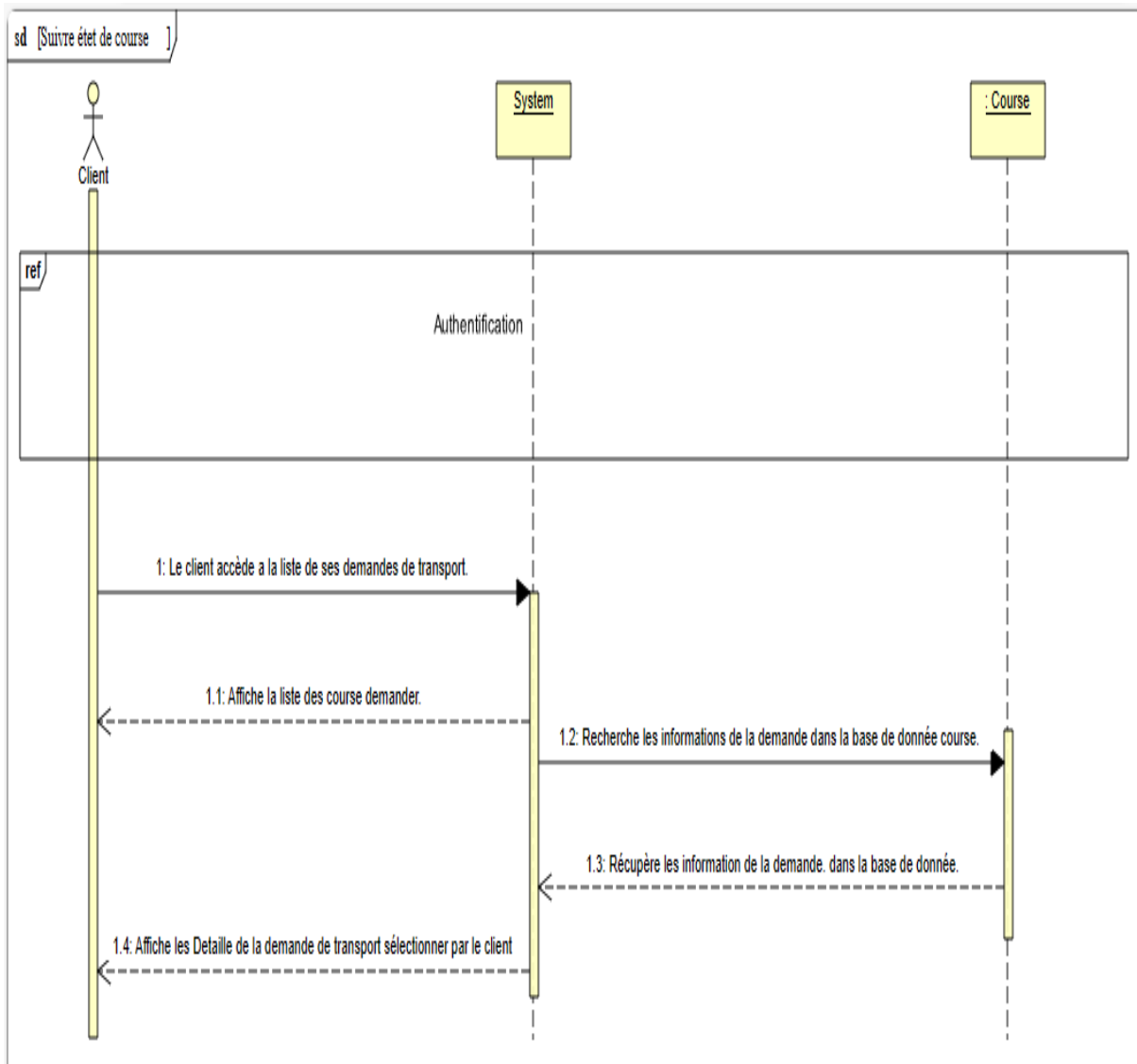


FIGURE 3.26 – Diagramme de séquence détaillé du cas «Suivre état de course».

3.4.5 Diagramme de classe

La Figure 3.27 suivante représente le diagramme de classes issu du Sprint 3.

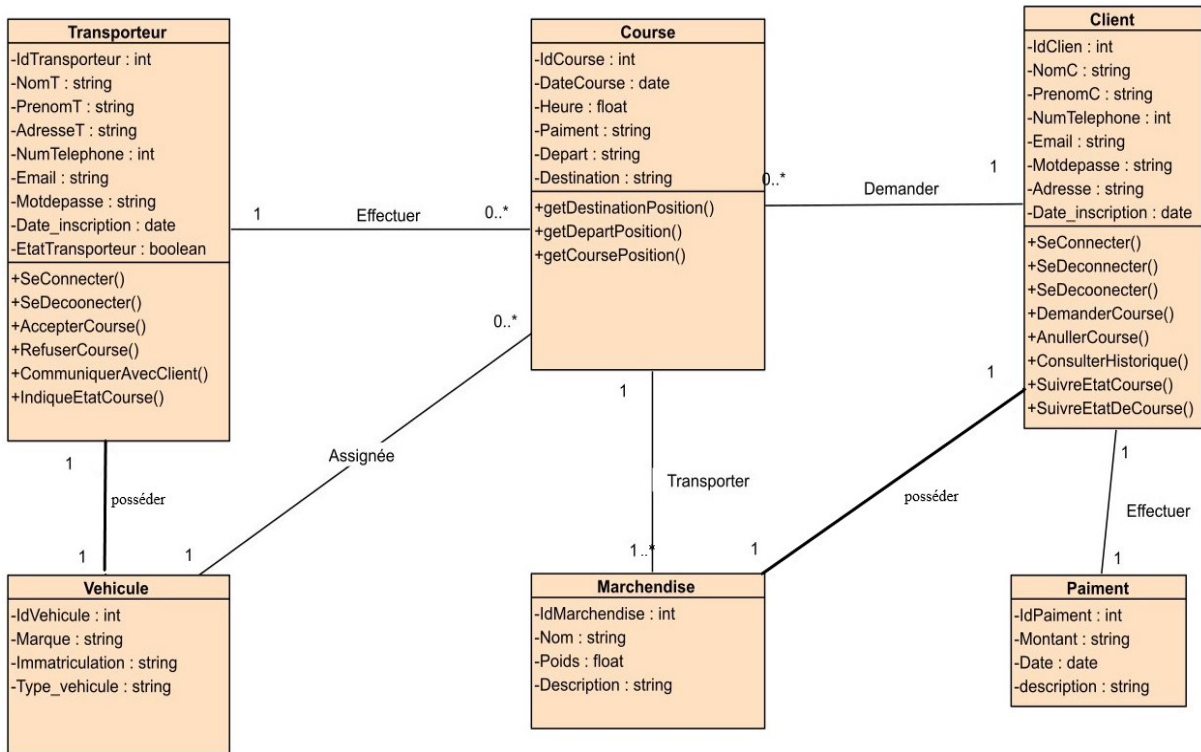


FIGURE 3.27 – Diagramme de classe du sprint 3.

3.4.6 Modèle relationnel

Voici le modèle relationnel correspondant au diagramme de classe du sprint 3

Transporteur(idTransporteur, nomT, prenomT, adresseT, numTelephone, email, motdepasse, date_inscription, etatTransporteur)

Vehicule(idVehicule, #idTransporteur, marque, immatriculation, type_vehicule)

Client(idClient, nomC, prenomC, numTelephone, email, motdepasse, adresse, date_inscription, #idPaiement)

Marchandise(idMarchandise, #idClient, nom, poids, description)

Course(idCourse, dateCourse, heure, paiement, depart, destination, #idVehicule, #idMarchandise, #idClient, #idTransporteur)

Paiement(idPaiement, montant, date, description)

3.4.7 Maquette

Cette interface présente de manière exhaustive la liste des courses du client, incluant l'état de chaque réservation, la date de la course, ainsi que des détails complets pour chaque trajet.

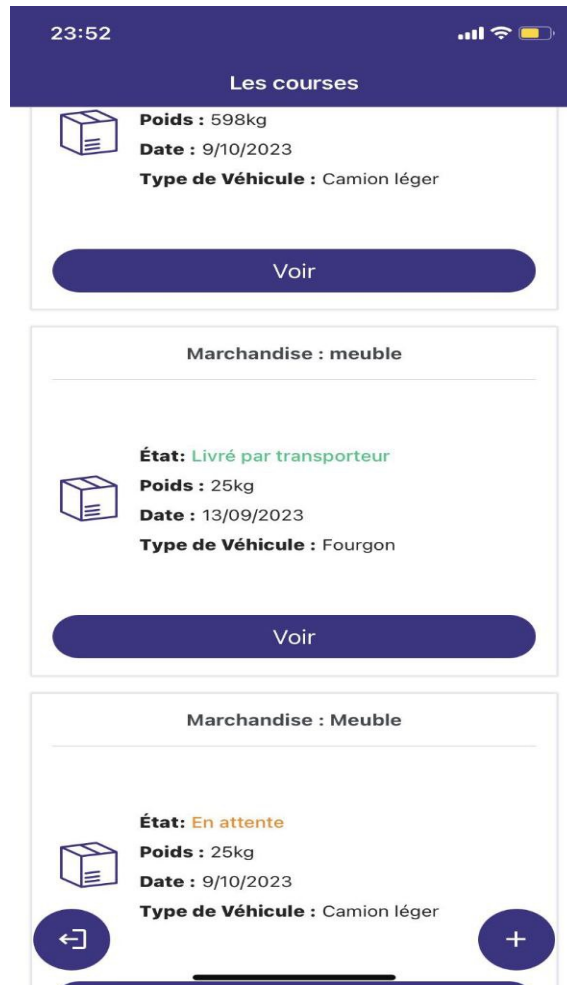


FIGURE 3.28 – Interface « liste des courses ».

Cette interface permet au transporteur de modifier l'état d'une course ou de l'annuler selon les besoins



22:26
◀ Camera

Les courses Marchandise

Point de Chargement :
Wilaya: Alger
Adresse: Béjaïa

Point de Livraison :
Wilaya: Batna
Adresse: Batna

Départ : 11/09/2023

Type de Véhicule : Fourgon

Poids de la marchandise: 50kg

Type de marchandise: Meuble

Nom et Prenom 1: Lounes

Numéro telephone: 0792015402

Prix: Négociable

Livré avec succès

Annulé

FIGURE 3.29 – Interface « modifier l'état d'une course ».

Dans cette interface, l'administrateur dispose d'un contrôle total sur les détails des marchandises, ainsi que sur les listes des transporteurs et des clients. Cela signifie qu'il peut accéder et gérer toutes les informations relatives aux marchandises en transit, superviser les transporteurs enregistrés, et avoir une vue complète sur la base de données des clients, leur permettant ainsi d'assurer une gestion efficace et centralisée de la plateforme

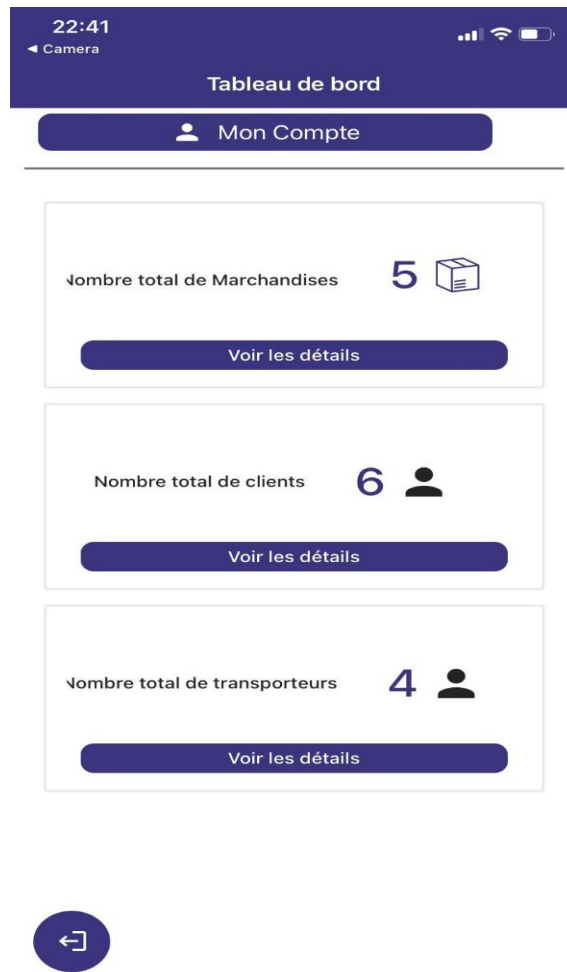


FIGURE 3.30 – Interface « tableau de bord de l'administrateur ».

3.5 Conclusion

En résumé, Ce chapitre nous a permis de concevoir notre application en tirant parti des trois sprints. Chaque sprint a enrichi notre compréhension des besoins des utilisateurs et des fonctionnalités essentielles. Avec une base solide établie, nous sommes maintenant prêts à passer à la phase de développement, confiants dans notre plan pour créer une application fonctionnelle. Dans le chapitre suivant, nous entrerons dans les détails de la réalisation de notre projet.

4.1 Introduction

Le chapitre de réalisation est le moment où nous mettons en pratique tout ce que nous avons planifié. Nous allons parler des outils et des logiciels que nous avons utilisés pour créer notre application "Wasla" de transport de marchandises en Algérie.

4.2 Environnement et outils de développement

4.2.1 Visual Paradigm

Est un outil de modélisation visuelle qui permet de concevoir et de modéliser des systèmes logiciels et des processus métier. Il fournit des fonctionnalités de diagramme UML, de modélisation de processus et d'ingénierie des exigences, entre autres [7].



FIGURE 4.1 – Logo Visual Paradigm.

4.2.2 Draw.io

Est un outil de création de diagrammes en ligne. Il permet de créer différents types de diagrammes tels que des diagrammes de flux, des organigrammes, des diagrammes UML, etc [11].

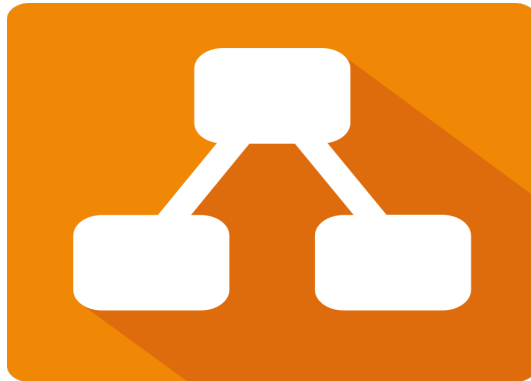


FIGURE 4.2 – Logo Draw.io.

4.2.3 Lucidchart

Est une plateforme de création de diagrammes en ligne qui permet de concevoir et de partager des diagrammes et des schémas. Il offre des fonctionnalités de collaboration en temps réel et prend en charge une variété de types de diagrammes[10].

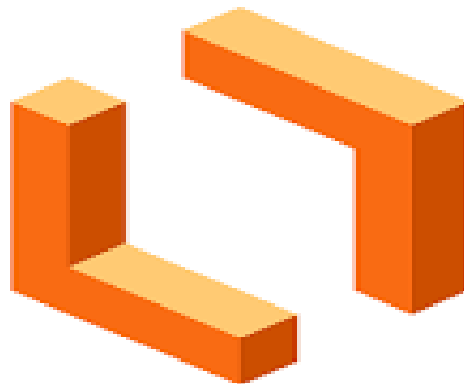


FIGURE 4.3 – Logo Lucidchart.

4.2.4 Visual Studio

Est un environnement de développement intégré (EDI) utilisé pour créer des applications pour diverses plates-formes telles que Windows, Android et iOS. Il prend en charge plusieurs

langages de programmation et offre des fonctionnalités de débogage, d'édition de code et de déploiement[6].



FIGURE 4.4 – Logo Visual Studio.

4.2.5 Android Studio

Android Studio est un environnement de développement intégré (EDI) utilisé pour créer des applications pour diverses plates-formes telles que Windows, Android et iOS. Il prend en charge plusieurs langages de programmation et offre des fonctionnalités de débogage, d'édition de code et de déploiement.



FIGURE 4.5 – Logo Android Studio.

4.2.6 Expo

Est un ensemble d'outils et de services qui simplifient de manière significative le processus de développement d'applications mobiles multiplateformes, en se concentrant principalement sur les applications pour iOS et Android. Il s'agit d'un framework open-source qui vise à réduire la complexité du développement d'applications en fournissant un ensemble complet de fonctionnalités préconfigurées, d'outils de développement et de services prêts à l'emploi[12].



FIGURE 4.6 – Logo Expo.

4.3 Front-end

4.3.1 React Native

React Native est un framework open-source développé par Facebook qui permet aux développeurs de créer des applications mobiles multiplateformes en utilisant principalement JavaScript et React. Il offre la possibilité de développer des applications mobiles natives pour iOS et Android en utilisant un seul code source, ce qui permet d'économiser du temps et des efforts en évitant le besoin de développer séparément pour chaque plateforme [9].

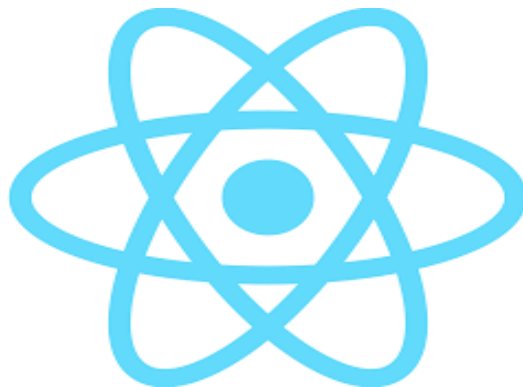


FIGURE 4.7 – Logo React Native.

4.4 Back-end

4.4.1 Node.js

Est un environnement d'exécution JavaScript côté serveur. Il permet de développer des applications Web évolutives et des services réseau en utilisant JavaScript. Node.js utilise une architecture orientée événements et basée sur des flux pour une efficacité maximale[13].

4.5 Implémentation de la base de données

4.5.1 NoSQL

Le terme NoSQL (Not Only SQL) fait référence à des types de bases de données non relationnelles, qui stockent les données dans un format différent des tables relationnelles. Cependant, les bases de données NoSQL peuvent être interrogées à l'aide d'API de langage idiomatique et de langages de requête structurés déclaratifs .

4.5.2 Fairebase

L'intégration de Fairebase dans notre application a été une étape importante pour assurer des fonctionnalités avancées de stockage et de gestion des données en temps réel. Fairebase, en tant que plateforme de développement mobile de Google, a apporté une solution efficace pour la gestion des utilisateurs, le stockage sécurisé des données, l'envoi de notifications push, et bien d'autres services essentiels. Son système de base de données en temps réel en particulier a permis d'améliorer la réactivité de notre application, en permettant aux utilisateurs de voir instantanément les mises à jour et les changements apportés par d'autres utilisateurs. De plus, Fairebase a facilité la mise en œuvre de l'authentification sécurisée et a contribué à la création d'une expérience utilisateur fluide et fiable. En résumé, l'intégration de Fairebase a grandement renforcé la robustesse et la performance de notre application, tout en simplifiant la gestion des données et en offrant des fonctionnalités avancées pour une expérience utilisateur exceptionnelle[14].

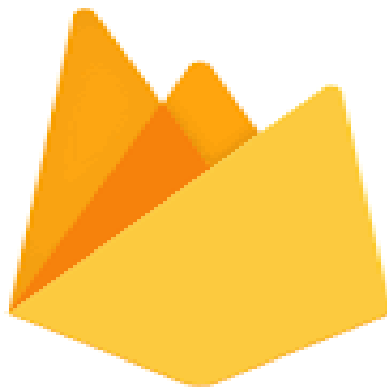


FIGURE 4.8 – Logo Fairebase.

4.6 Conclusion

Dans ce chapitre, nous avons présenté les outils et les technologies utilisés pour le développement de l'application mobile "Wasla" pour le transport de marchandises en Algérie. L'ap-

plication a été développée en utilisant un ensemble d'outils et de technologies open-source, notamment React Native, Node.js et Firebase. Ces outils et technologies ont été choisis en fonction des besoins spécifiques de l'application.

CONCLUSION GÉNÉRALE ET PERSPECTIVES

Ce mémoire nous a permis d'explorer en profondeur le domaine du transport de marchandises à l'ère de la numérisation, et nous avons réussi à concevoir une application mobile dédiée à la demande de services de transport, "Wasla". Nous avons abordé en détail la problématique complexe consistant à simplifier et à optimiser le processus de demande de transport de marchandises. Grâce à notre recherche approfondie et à la méthodologie de développement scrum, nous avons démontré que "Wasla" offre une solution viable à cette problématique. L'application est dotée d'une interface conviviale, améliore la traçabilité des marchandises et favorise une communication transparente entre toutes les parties prenantes, ce qui contribue à accroître l'efficacité opérationnelle et la satisfaction globale des utilisateurs. L'innovation a toujours été au cœur de notre démarche, et "Wasla" a le potentiel de transformer le secteur du transport de marchandises en intégrant la technologie au cœur de ses opérations. En proposant une solution qui simplifie et optimise le processus de demande, nous contribuons activement à l'innovation de l'industrie tout en répondant aux besoins changeants des entreprises et des transporteurs.

Nous nous efforçons d'atteindre la première version complète de l'application, en nous assurant que toutes les fonctionnalités essentielles sont opérationnelles. Nous entreprendrons des tests rigoureux de l'application pour garantir sa stabilité et son efficacité. Nous solliciterons activement les retours d'information des utilisateurs, y compris des bêta-testeurs, afin d'identifier et de corriger tout problème potentiel.

Notre stratégie de lancement sera soigneusement planifiée, en commençant par une approche ciblée. Nous envisageons de déployer "Wasla" dans une ou quelques wilayas spécifiques, où nous pourrions consolider notre présence et évaluer la réception de l'application.

Nous mettrons en place une stratégie de marketing solide pour attirer les utilisateurs et construire une base solide de clients. Cela comprendra la création de sites web, de comptes de médias sociaux et la planification de campagnes publicitaires.

En plus des perspectives mentionnées précédemment, nous prévoyons également d'intégrer

des fonctionnalités de paiement en ligne, ce qui permettra aux utilisateurs de régler leurs transactions de manière électronique, offrant ainsi une expérience utilisateur encore plus fluide et sécurisée. Cette fonctionnalité de paiement en ligne contribuera à la croissance de notre application en simplifiant davantage le processus pour les clients et les transporteurs.

Nous travaillons sur un modèle économique pour garantir une croissance rentable et durable de notre startup. Nous explorerons des options telles que des abonnements premium ou d'autres sources de revenus potentielles.

Nous prévoyons des mises à jour régulières de l'application pour rester compétitifs et intégrer de nouvelles fonctionnalités. L'innovation continue restera au cœur de notre démarche pour maintenir notre avantage concurrentiel.

En conclusion, notre mémoire est un jalon majeur dans la création de notre startup. "Wasla" est prête à transformer le secteur du transport de marchandises en Algérie grâce à son innovation et à sa technologie de pointe. Avec une planification méticuleuse et une exécution solide de ces perspectives, nous sommes convaincus que notre startup atteindra de nouveaux sommets et façonnera l'avenir du transport de marchandises.

BIBLIOGRAPHIE

- [1] J.-P. Subra and A. Vannieuwenhuyze, *Scrum : Une méthode agile pour vos projets*. Éditions ENI., 2019.
- [2] P. Deemer, G. Benefield, C. Larman, and B. Vodde, *Guide Léger de la Théorie et de la Pratique de Scrum, version 2.0*. Diateino, 2009.
- [3] K. Schwaber and J. Sutherland, “The scrum guide,” *Scrum Alliance*, vol. 21, no. 1, pp. 1–38, 2011.
- [4] F. Debois and J. Maes, *La boîte à outils du chef de projet*. Dunod, 2013.
- [5] T. Pigot, “Scrum en moins de 10 minutes,” année de publication.
- [6] D. Flanagan, *JavaScript : das umfassende Referenzwerk ;[behandelt Ajax und DOM scripting]*. O’Reilly Germany, 2007.
- [7] G. Booch, J. Rumbaugh, I. Jacobson, *et al.*, *Le guide de l’utilisateur UML*, vol. 3. Eyrolles, 2000.
- [8] S. Chaabna and H. Wang, “Analysis of the state of e-commerce in algeria,” *International Journal of Marketing Studies*, vol. 7, no. 2, p. 44, 2015.
- [9] “<https://reactnative.dev/>,” Visiter le 12/05/2023.
- [10] “<https://www.lucidchart.com/>,” Visiter le 21/04/2023.
- [11] “<https://app.diagrams.net/>,” Visiter le 02/07/2023.
- [12] “<https://docs.expo.dev/>,” Visiter le 19/07/2023.
- [13] “<https://nodejs.org/fr/>,” Visiter le 12/06/2023.
- [14] “<https://console.firebase.google.com/>,” Visiter le 15/08/2023.
- [15] A. OUZZANI, O. TIJANI, and A. REKLAOUI, “L’impact du canal de suez sur l’évolution de l’économie mondiale,” *Revue des Etudes et Recherche en Logistique et Développement*, vol. 3, pp. 55–78, 2018.

RÉSUMÉ

Dans le cadre de ce projet de fin de cycle, l'objectif principal était de créer la startup "Wasla", une application mobile permettant de mettre en relation les clients, qu'ils soient des entreprises ou des particuliers, avec des transporteurs pour le transport de marchandises. Pour atteindre cet objectif, nous avons adopté une approche agile, plus précisément Scrum, une méthode de gestion de projet itérative et incrémentale. La conception de l'application a été réalisée en utilisant UML comme langage de modélisation. L'application a été développée dans l'environnement de développement intégré (IDE) Visual Studio Code, en utilisant les technologies React Native et Node.js pour la programmation. Elle permet aux clients de trouver des transporteurs adaptés à leurs besoins en matière de transport, tout en offrant aux transporteurs la possibilité de trouver et de communiquer avec les clients. "Wasla" vise ainsi à simplifier et faciliter le processus de mise en relation entre les clients et les transporteurs pour le transport de marchandises, répondant ainsi aux besoins évolutifs du secteur du transport de marchandises.

Mots clés : Scrum, UML, React Native, transport, marchandise, Wasla.

ABSTRACT

As part of this end-of-cycle project, the main objective was to create the startup "Wasla," a mobile application that connects customers, whether they are businesses or individuals, with transporters for the transportation of goods. To achieve this goal, we adopted an agile approach, specifically Scrum, an iterative and incremental project management method. The application's design was executed using UML as the modeling language. The application was developed in the Visual Studio Code integrated development environment (IDE), using React Native and Node.js technologies for programming. It allows customers to find transporters tailored to their transportation needs, while also providing transporters with the ability to find and communicate with customers. "Wasla" aims to simplify and streamline the matchmaking process between customers and transporters for the transportation of goods, thus meeting the evolving needs of the freight transport sector.

Key words: Scrum, UML, React Native, transportation, merchandise, Wasla.