

République Tunisienne
Ministère de l'Enseignement Supérieur et de
la Recherche Scientifique
Université de Sousse



Institut Supérieur des Sciences
Appliquées et de Technologie de Sousse



DEPARTEMENT INFORMATIQUE

RAPPORT DE PROJET DE FIN D'ÉTUDES

En vue de l'obtention du :

*Diplôme National d'Ingénieur Informatique
Option : Génie Logiciel-Architecture Logicielle*

Conception et Réalisation d'une application mobile pour lutter contre le gaspillage alimentaire

Elaboré par : Maha KCHAICH

Soutenu le 07/07/2023 devant le jury :

<i>Président :</i>	Dr. BLAIECH Ahmed Ghazi	ISSAT Sousse
<i>Examinateur :</i>	Dr. ZARROUK Elyes	ISSAT Sousse
<i>Encadrant :</i>	Dr. CHTIOUI Houssem	ISSAT Sousse
<i>Encadrant Industriel :</i>	Ing. KNANI Mariem	Webify

Année Universitaire : 2022-2023

Code Sujet : FI-GL23-029

Dédicace

Je dédie ce travail à mon très cher père, **Mohamed**, et à ma très chère mère, **Moufida**. Je ne saurais trouver les mots pour exprimer vos mérites et tous les sacrifices que vous avez consentis au détriment de votre propre confort depuis le jour de ma naissance jusqu'à aujourd'hui. Votre dévouement, votre affection et votre soutien inconditionnel m'ont toujours encouragée à faire les meilleurs choix, que ce soit dans ma vie quotidienne ou dans mes études, et à donner le meilleur de moi-même pour vous rendre fiers de votre fille.

À mes chères sœurs, **Mariem**, **Marwa**, **Manar**, et mon frère, **Mohamed**, je ne peux trouver les mots justes pour exprimer mon affection et mes pensées. Vous êtes pour moi les personnes sur qui je peux compter inconditionnellement. Que Dieu tout-puissant vous préserve et vous offre santé et bonheur.

À mon cher **Chiheb**,
je suis vraiment **bouleversée** par l'immense gratitude que je ressens envers toi pour le soutien infaillible que tu m'as toujours accordé. Les mots me manquent pour exprimer à quel point je suis reconnaissante. Depuis le tout début, tu as été là pour moi, m'encourageant sans relâche et me poussant à donner le meilleur de moi-même. Tu es une source constante d'amour et d'inspiration, et je suis reconnaissante d'avoir la chance de t'avoir dans ma vie.

Je souhaite également dédier ces lignes
à mes neveux, **Moussa**, **Younes**, **Aziz**, **Mohamed** et à ma nièce, **Manessa**, qui apportent joie et bonheur dans nos vies.

À mes chers beaux-frères, **Saïd** et **Ahmed**, ainsi qu'à ma belle-sœur, **Naoures**, qui n'ont jamais hésité à m'aider.

À mes chers amis les plus proches ^à de mon cœur,
Siwar, **Dhekra**, **Rihem**, **Rami**, **Saber**, **Farah**, **Wifek**, **Aslene**, **Roua** et **Ahmed**, je tiens à exprimer toute ma gratitude et ma reconnaissance pour votre soutien indéfectible. Vous avez toujours été là pour moi, prêts à m'aider dans les moments difficiles et à partager les moments de bonheur. Vous êtes les personnes qui ont marqué mes plus beaux souvenirs et qui ont contribué à rendre ma vie plus riche et plus significative.

À tous ceux qui m'aiment. Pour vous, je dédie ce travail modeste.

Remerciements

Ma sincère gratitude et mes profonds remerciements vont à Dr Houssem CHTIOUI, mon encadrant à Institut Supérieur des Sciences Appliquées et de Technologie de Sousse, pour son suivi attentif et minutieux, sa patience, ses conseils pertinents et l'attention qu'il m'a accordée tout au long de mon travail.

Accepter de m'encadrer pour ce PFE est une faveur dont je suis très fière et reconnaissante.

Je remercie également M. Islem DAHMANE, Mme Mariem KNANI, M. Achref TALEB, notre encadrante professionnelle, ainsi que le personnel administratif et technique de la société Webify pour leur chaleureux accueil, leurs précieux conseils, les informations fournies et la motivation qu'ils m'ont apportés, ce qui m'a permis de persévérer jusqu'au bout pour réussir ce projet.

Je remercie également l'ensemble du personnel administratif ainsi que les enseignants de l'Institut Supérieur des Sciences Appliquées et de Technologie de Sousse pour la qualité de la formation dispensée et l'ambiance agréable qu'ils ont eu instaurer tout au long de mes années d'études.

Je présente ma profonde gratitude aux membres du jury qui me font l'immense honneur d'accepter de juger ce travail, et je les remercie pour ce privilège.

Enfin, que tous ceux et celles qui ont contribué de près ou de loin à l'accomplissement de ce travail trouvent ici l'expression de nos remerciements les plus chaleureux.

mes

Résumé

Ce travail a été élaboré dans le cadre du projet de fin d'études pour l'obtention du diplôme national d'ingénieur en informatique, qui a été réalisé au sein de la société WEBIFY. Notre projet consiste à analyser, concevoir et réaliser une application mobile "SAVERFOOD", dédiée pour lutter contre le gaspillage alimentaire.

Mots clés : Flutter, Model-View-Controller, Dart, Laravel, Angular, SCRUM.

Abstract

This work was developed as part of the final year project to obtain National Diploma in Computer Engineering, which was carried out within the company WEBIFY. Our project consists of analyzing, designing, and implementing a mobile application called "SAVERFOOD," dedicated **reducing** to **combating** food waste.

keywords : Flutter, Model-View-Controller, Dart, Laravel, Angular, SCRUM.

Table des matières

Table des figures	vi
Liste des tableaux	viii
Introduction générale	1
1 Cadre Général du projet	3
Introduction	3
1.1 Présentation de l'organisme d'accueil	3
1.1.1 Présentation	3
1.1.2 Services	3
1.1.3 Coordonnées	4
1.2 Présentation du projet	4
1.2.1 Problématique	4
1.2.2 Etude de l'existant	5
1.2.3 Solution proposée	7
1.2.4 Approche basée sur l'intelligence artificielle	7
1.3 Méthodologie de gestion du projet	8
1.3.1 Méthode agile	8
1.3.2 SCRUM	8
1.3.3 Méthodes Agiles : SCRUM (Rôles)	8
1.3.4 Méthodes Agiles : SCRUM (Pratiques)	9
1.4 Planification du projet	10
Conclusion	10
2 Analyse et spécification des besoins	11
Introduction	11
2.1 Identification des acteurs	11
2.1.1 Partie Web (Back-office)	11
2.1.2 Partie Mobile	11
2.2 Spécification des besoins	12
2.2.1 Les besoins fonctionnels	12
2.2.2 Les besoins non fonctionnels	14
2.3 Modélisation des besoins	15
2.3.1 Langage de modélisation UML	15
2.3.2 Diagramme de cas d'utilisation Global de la partie Web (Back-office) . .	15
2.3.3 Diagramme de cas d'utilisation Global de la partie mobile	16

2.4	Modélisation fonctionnelle des fonctionnalités système	16
2.4.1	Sprint 1 : Partie Web	16
2.4.2	Sprint 2 : Partie mobile côté partenaire	24
2.4.3	Sprint 3 : Partie mobile côté client	34
	Conclusion	43
3	Aperçu conceptuel	44
	Introduction	44
3.1	Architecture générale	44
3.1.1	Les patrons d'architecture	44
3.1.2	Les patrons de conception	46
3.2	Conception détaillée	47
3.2.1	Diagramme de classes des entités	47
3.2.2	Conception détaillée de sprint 1	49
3.2.3	Conception détaillée de sprint 2	52
3.2.4	Conception détaillée de sprint 3	58
	Conclusion	60
4	Réalisation	61
	Introduction	61
4.1	Spécification technique	61
4.1.1	Environnement matériel	61
4.1.2	Environnement logiciel	61
4.2	Diagramme de déploiement	67
4.3	Réalisation	67
4.3.1	Réalisation du sprint 1 : Interfaces des administrateurs	67
4.3.2	Réalisation du sprint 2 : Interfaces des partenaires	72
4.3.3	Réalisation du sprint 3 : Interfaces des clients	76
4.3.4	Réalisation du sprint 4	80
	Conclusion	88
	Conclusion générale et perspectives	89
	Références	91

Table des figures

1.1	Les interfaces de l'application mobile TooGoodToGo	5
1.2	Les interfaces de l'application mobile Flashfood	6
1.3	Les interfaces de l'application mobile Olio	6
1.4	Répartition de Sprints	9
1.5	Diagramme de GANTT	10
2.1	Diagramme de cas d'utilisation générale du partie Web	15
2.2	Diagramme de cas d'utilisation générale du partie mobile	16
2.3	Diagramme de navigation de partie Web	de la partie web 18
2.4	Diagramme détaillé de cas d'utilisation « Gérer les partenaires »	19
2.5	Diagramme détaillé de cas d'utilisation « Gérer les clients »	21
2.6	Diagramme détaillé de cas d'utilisation « Gérer les boxes »	23
2.7	Diagramme de navigation de partie mobile côté partenaire	de la partie 26
2.8	Diagramme de séquence système « S'authentifier»	27
2.9	Diagramme de séquence système « Gérer profil »	29
2.10	Diagramme de séquence système « Gérer boxes »	31
2.11	Diagramme de séquence système « Consulter commandes »	33
2.12	Diagramme de navigation de partie mobile côté client	de la partie 36
2.13	Diagramme détaillé de cas d'utilisation « Réserver box »	37
2.14	Diagramme détaillé de cas d'utilisation « Valider commandes »	38
2.15	Diagramme détaillé de cas d'utilisation « Sélectionner boxes»	40
2.16	Diagramme détaillé de cas d'utilisation « Gérer les favoris »	42
3.1	Architecture 3-tiers	45
3.2	Principe du patron MVC	46
3.3	Le patron de conception observer	47
3.4	Diagramme de classes des entités	48
3.5	DS du cas d'utilisation « Gérer les partenaires »	50
3.6	DS du cas d'utilisation « Gérer les clients »	51
3.7	DS du cas d'utilisation « Gérer les boxes »	52
3.8	DCP du cas d'utilisation « S'authentifier »	53
3.9	DCP du cas d'utilisation « Gérer profil »	54
3.10	DCP du cas d'utilisation « Gérer boxes »	55
3.11	Diagramme d'états-transitions du cas d'utilisation « Gérer boxes »	56
3.12	DCP du cas d'utilisation « Consulter commandes »	56
3.13	DCC du cas d'utilisation « Consulter commandes »	57
3.14	DS du cas d'utilisation « Réserver Box »	58

3.15 DS du cas d'utilisation « Sélectionner boxes »	59
3.16 DS du cas d'utilisation « Valider Commandes »	60
4.1 Diagramme de déploiement	67
4.2 L'interface d'authentification pour l'administrateur	68
4.3 L'interface de tableau de bord	69
4.4 L'interface de gestion des partenaires	69
4.5 L'interface d'ajout de client	70
4.6 L'interfaces de gestion des boxes	71
4.7 L'interface de détails d'une commande <i>des détails</i>	71
4.8 L'interface d'autorisation	72
4.9 Les interfaces d'accueil	73
4.10 Les interfaces d'inscription de partenaire	73
4.11 L'interface d'authentification	74
4.12 Les interfaces de gestion des commandes	75
4.13 Les interfaces de gestion des boxes	75
4.14 Les interfaces de gestion de profil	76
4.15 Les interfaces de récupération de mot de passe	76
4.16 Les interfaces de consultation d'accueil	77
4.17 Les interfaces pour passer commande	78
4.18 Les interfaces de confirmation de commande	78
4.19 Les interfaces de sélection de boxes	79
4.20 Les interfaces de consultation des favoris	79
4.21 Les interfaces de gestion de profil	80
4.22 Flux de travail	81
4.23 Données de l'application en ligne ThunderExpress	82
4.24 Données initial	82
4.25 Colonnes supprimées du jeu de données	83
4.26 Colonnes conservées du jeu de données	83
4.27 Aggrégation par colonne	84
4.28 Conversion en vecteurs	84
4.29 L'algorithme similarité cosinus	85
4.30 Mesure de précision	85
4.31 Processus d'intégration de système de recommandation	86
4.32 Graphe du système créé avec Neo4j	87
4.33 Graphe du système créé avec Neo4j	88

Liste des tableaux

1.1	Étude de l'existant	7
1.2	Acteurs du projet selon "Scrum"	8
2.1	Backlog du produit de partie web	17
2.2	Cas d'utilisation « Gérer les partenaires »	21
2.3	Description textuelle du cas « Gérer les clients »	22
2.4	Description textuelle du cas «Gérer les boxes»	24
2.5	Backlog du produit de partie mobile côté partenaire	25
2.6	Description textuelle du cas « S'authentifier »	28
2.7	Description textuelle du cas « Gérer profil »	30
2.8	Description textuelle du cas «Archiver boxes»	32
2.9	Description textuelle du cas « Consulter commandes »	34
2.10	Backlog du produit de partie mobile côté client	35
2.11	Description textuelle du cas « Réserver box »	38
2.12	Description textuelle du cas « Valider commandes »	39
2.13	Description textuelle du cas « Sélectionner boxes»	41
2.14	Description textuelle du cas « Gérer les favoris »	43
3.1	Description des classes	49
4.1	Matériel de base	61

Introduction générale

Le gaspillage alimentaire est un problème mondial préoccupant qui a des conséquences profondes sur l'environnement, l'économie et la société dans son ensemble. Chaque année, des quantités massives de nourriture sont jetées, alors même que des millions de personnes souffrent de la faim à travers le monde.

Le gaspillage alimentaire se produit à toutes les étapes de la chaîne d'approvisionnement alimentaire, depuis la production agricole jusqu'à la consommation finale. Les pertes peuvent survenir en raison de la surproduction, de la mauvaise gestion des stocks, des dates de péremption, ainsi que des habitudes de consommation et des comportements de gaspillage.

Selon les estimations, environ un tiers de la production alimentaire mondiale est gaspillée chaque année, ce qui équivaut à des milliards de tonnes de nourriture. Cela a un impact environnemental significatif, car la production alimentaire nécessite des ressources précieuses telles que l'eau, les terres agricoles et l'énergie.

Face à ce problème majeur, de nombreuses initiatives ont été lancées pour lutter contre le gaspillage alimentaire. Des organisations, des gouvernements, des entreprises et des particuliers s'engagent à réduire les pertes et à valoriser les aliments non consommés. Les technologies, telles que les applications mobiles, sont également utilisées pour faciliter la récupération et la redistribution des invendus.

Dans ce contexte, notre projet de fin d'études vise à contribuer à la lutte contre le gaspillage alimentaire en développant une application. En sensibilisant les utilisateurs et en offrant des solutions pratiques pour récupérer les invendus à prix réduit, nous souhaitons apporter ma notre contribution à la réduction du gaspillage alimentaire et à la construction d'un avenir plus durable.

Le présent rapport s'articule autour de quatre chapitres :

Le premier chapitre, intitulé “**Cadre Général du projet**”, constitue une introduction de l'organisme d'accueil, puis nous passons à l'étude de l'existant, ainsi que la solution proposée. Nous terminerons par la description de méthodologie de gestion du projet adoptée.

Le deuxième chapitre, intitulé “**Analyse et spécification des besoins**”, définit les acteurs de notre application et spécifie les besoins fonctionnels et non fonctionnels auxquels notre application doit répondre, tout en présentant ses principales fonctionnalités.

Le troisième chapitre, intitulé “**Aperçu conceptuel**”, décrit les schémas conceptuels et l'ar-

chitecture adoptée pour la solution proposée, avec une explication du comportement dynamique de l'application.

Le quatrième chapitre, intitulé "**Réalisation**", abordera en détail l'environnement de développement matériel et logiciel, ainsi que le travail réalisé dans notre projet.

Finalement, le rapport est clôturé par une conclusion générale qui résume notre travail et nous proposons quelques perspectives.

Cadre Général du projet

Introduction

Dans ce premier chapitre, nous allons spécifier le cadre général de notre projet. Tout d'abord, nous allons présenter l'organisme d'accueil. Ensuite, nous allons détailler la problématique de notre stage ainsi que la solution proposée, qui consistera en une tentative de résoudre les inconvénients des solutions existantes. Nous allons terminer ce chapitre en décrivant le calendrier de notre stage à travers le diagramme de Gantt, et nous allons spécifier également la méthodologie de gestion du projet.

1.1 Présentation de l'organisme d'accueil

1.1.1 Présentation

WEBIFY TECHNOLOGY est une agence tunisienne fondée en 2019 à Sousse, formée d'une équipe d'ingénieurs doués et ardents. Cette société est spécialisée dans la conception et la réalisation de sites / applications Web et Mobiles. Elle est ouverte aux dernières tendances, afin d'offrir un service adéquat, suite à des connaissances diverses et un engagement à long terme. L'innovation est la valeur sûre de WEBIFY TECHNOLOGY, toute l'équipe aime tout ce qu'elle fait, la satisfaction et le respect de l'engagement est leur point fort. des engagements

L'approche d'affaires adoptée par WEBIFY TECHNOLOGY est axée sur l'écoute et les besoins de ses clients et partenaires dans le but de : fournir un conseil personnalisé bien déterminé, aider à atteindre les objectifs tout en respectant le budget ainsi que l'échéance prévu et accompagner, superviser, conseiller et épauler les clients.

1.1.2 Services

Les services proposés par Webify sont organisés, essentiellement selon trois axes à savoir :

- Management : Gestion de projets, Stratégies, Développement d'affaires à l'international.
- Technologies innovantes : Référencement et Web analytic, Marketing digital, Web responsive, Métadonnées, Géolocalisation, Automatisation.

des besoins

- Marketing interactif : Stratégie marketing, Identité visuelle, Conception et design emballage, Conception et design graphique, Production de tout type de contenu multimédia, Recherche et veille stratégique, Communications interactives, Campagnes de publicité et accompagnement marketing, Marketing par Emailing.

1.1.3 Coordonnées

Les coordonnées de l'entreprise sont détaillées ci-dessous :

Webify Tecchnology a deux branches, une à Sousse et l'autre à France.

Adresse : Immeuble jaziri Boulevard 14 janvier, HammamSousse

Téléphone : +216 25 23 43 69

LinkedIn : <https://www.linkedin.com/company/webify-technology/>

Facebook : <https://www.facebook.com/Webify.Tech>

Ce travail fait partie du projet de fin d'études visant à obtenir le diplôme d'ingénieur en génie logiciel de l'Institut Supérieur des Sciences Appliquées et de Technologie de Sousse. Il a été réalisé au sein de la startup Webify.



Logo de Webify

1.2 Présentation du projet

Dans cette section, nous mettons notre travail dans son contexte général. Tout d'abord, nous abordons la problématique qui nous a conduit à sélectionner ce sujet, ainsi que les raisons sous-jacentes. Ensuite, nous exposons la solution proposée afin de résoudre la problématique. À la fin, nous détaillons les objectifs de ce projet.

1.2.1 Problématique

Le gaspillage alimentaire constitue un véritable fléau de notre société, engendrant la perte et le rejet de quantités importantes de nourriture destinée à la consommation humaine. Ce phénomène préoccupant concerne de nombreux acteurs tout au long de la chaîne alimentaire, depuis la production jusqu'à la consommation. Chaque année, des millions de tonnes de denrées alimentaires sont ainsi jetées, alors même que des populations souffrent de la faim à travers le monde.

Ce gaspillage a des conséquences néfastes tant sur l'environnement que sur l'économie. En effet, il entraîne la perte des ressources précieuses utilisées pour produire ces aliments, contribuant ainsi à une utilisation inefficace et inutile des terres, de l'eau, de l'énergie et d'autres intrants agricoles.

Par ailleurs, ce gaspillage a un impact financier considérable, tant pour les consommateurs que pour les commerçants. De plus, les commerçants subissent des pertes financières importantes en raison de la destruction de produits encore parfaitement consommables.

Il est donc essentiel de prendre des mesures concrètes pour lutter contre le gaspillage alimentaire, en promouvant notamment la sensibilisation et l'exploit des excédents alimentaires.

1.2.2 Etude de l'existant

La phase d'étude de l'existant est cruciale. Elle consiste d'une part à analyser les applications similaires existantes dans le but d'identifier leurs points faibles pour les améliorer. D'autre part, elle facilite la spécification d'une nouvelle solution optimale et efficace.

Dans ce paragraphe, nous présentons les différentes solutions existantes étudiées dans le cadre de notre projet.

1. **Too Good To Go** : est une application mobile qui met en relation ses utilisateurs avec des boulangeries, restaurants, supermarchés, etc...afin de leur proposer des invendus à prix réduits sous la forme de paniers à sauver.

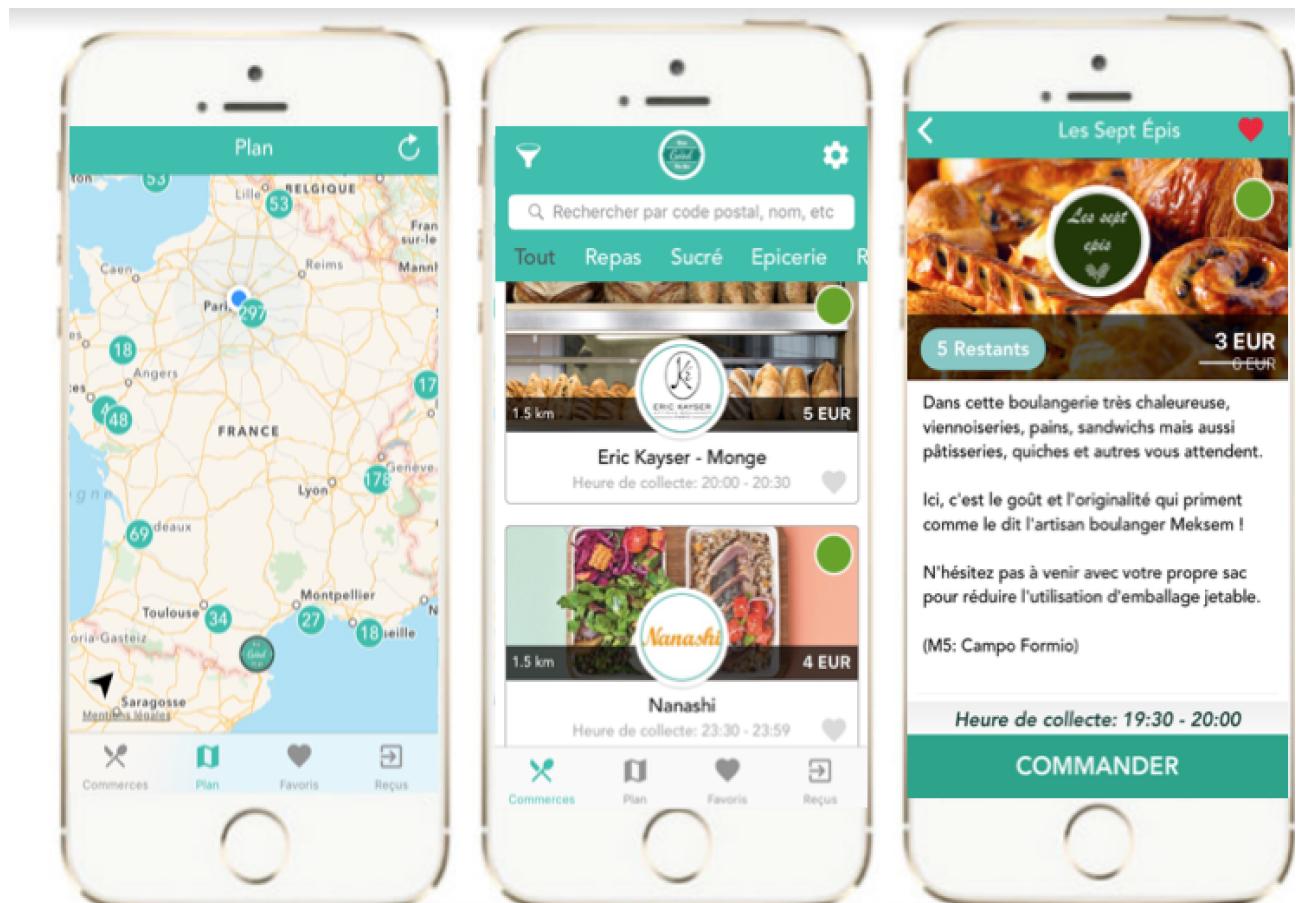


FIGURE 1.1 – Les interfaces de l'application mobile TooGoodToGo

2. **Flashfood** : est une application qui permet aux utilisateurs de commander des aliments à des prix réduits chez les supermarchés locaux.

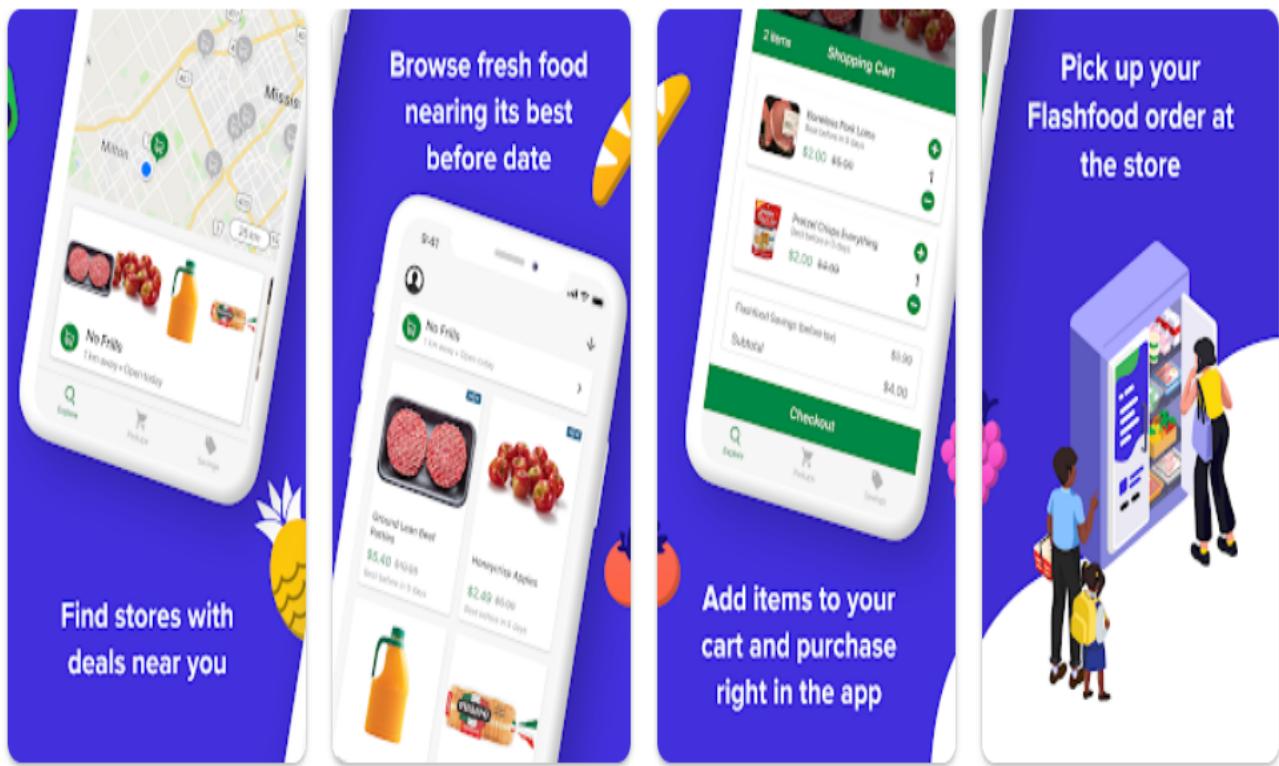


FIGURE 1.2 – Les interfaces de l’application mobile Flashfood

3. **Olio** : est une application qui permet aux utilisateurs de partager leurs excédents alimentaires avec leurs voisins plutôt que de les jeter.

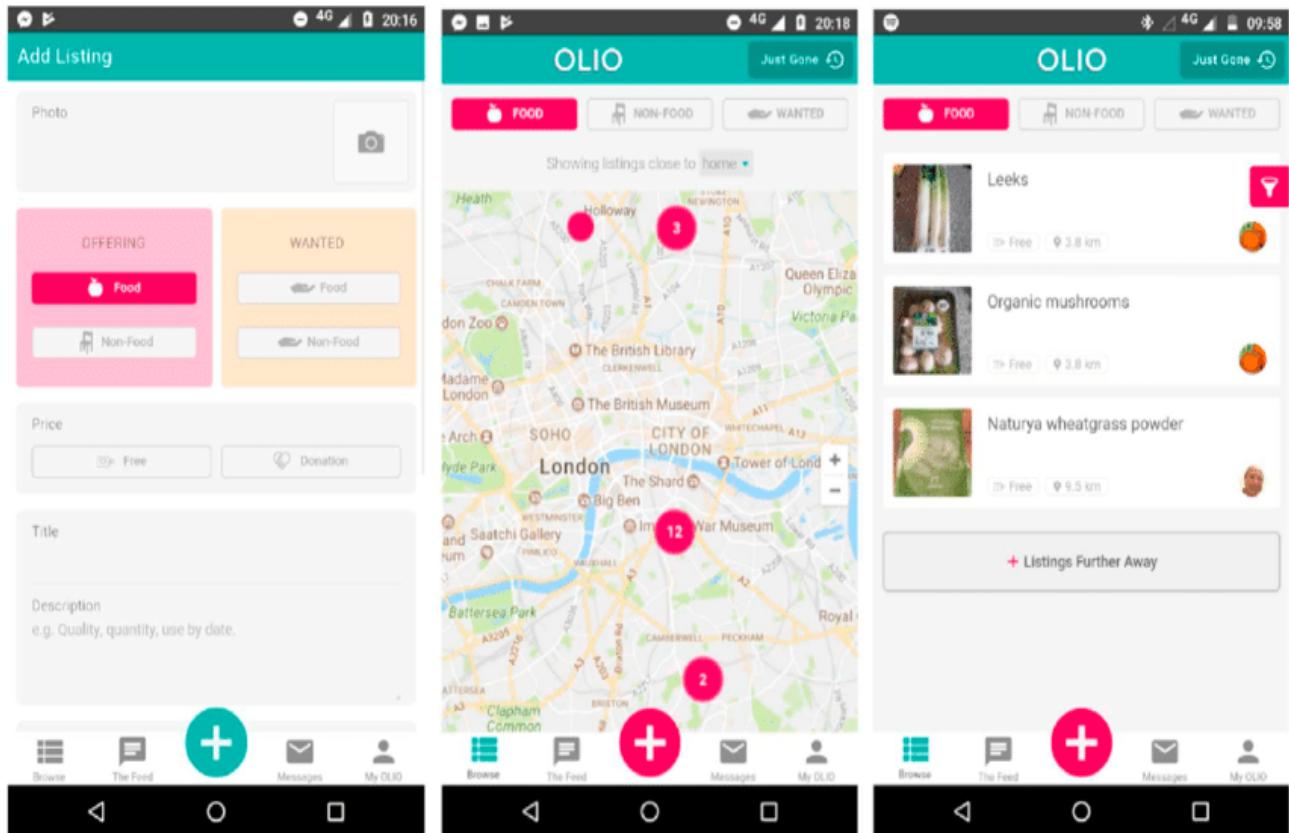


FIGURE 1.3 – Les interfaces de l’application mobile Olio

Nous analysons et critiquons ces applications à travers le tableau 1.1 :

Applications	Avantages	Inconvénients
Too Good To Go	<ul style="list-style-type: none"> Diversité des choix alimentaires en offrant aux utilisateurs la possibilité de découvrir de nouveaux restaurants, boulangeries ... Sensibilisation à la problématique du gaspillage en informant les utilisateurs sur la quantité d'aliments sauvés grâce à leurs achats. 	<ul style="list-style-type: none"> Les repas peuvent ne pas être toujours disponibles ou correspondre aux préférences des utilisateurs. L'application n'est pas disponible en Tunisie.
Flashfood	<ul style="list-style-type: none"> Promotion de la communauté locale en mettant en avant les commerçants et les producteurs locaux, ce qui peut encourager les achats locaux et soutenir l'économie locale. 	<ul style="list-style-type: none"> Cette application présente des produits de supermarchés locaux seulement. L'application n'est pas disponible en Tunisie.
Olio	<ul style="list-style-type: none"> Encourageant le partage et l'échange des aliments, renforçant ainsi les liens sociaux et la solidarité. Favorise la diversité alimentaire en permettant aux utilisateurs de découvrir de nouveaux aliments et recettes. 	<ul style="list-style-type: none"> Les partenaires de l'application sont tous des simples utilisateurs(C2C). L'application n'est pas disponible en Tunisie.

TABLE 1.1 – Étude de l'existant

L'objectif de notre application est d'intégrer tous les points forts des solutions existantes et en même temps compenser leurs points faibles.

1.2.3 Solution proposée

[partie précédente](#)

En tenant compte de la [partie d'étude de l'existant](#), nous proposons une application qui lutte contre le gaspillage alimentaire en connectant commerçants et clients, afin de sauver leurs invendus de la poubelle à la fermeture. Géolocalisés, les clients repèrent les partenaires commerçants autour d'eux, paient un petit prix, et passent pour récupérer leur « portion surprise », composée en fonction des invendus du jour.

Un concept gagnant-gagnant : le client profite de bons repas à prix réduits et s'engage au quotidien contre le gaspillage, le commerçant ne jette plus ses produits et rentre dans ses frais.

1.2.4 Approche basée sur l'intelligence artificielle

Dans notre solution, nous avons pris la décision d'intégrer un modèle intelligent reposant sur l'algorithme de similarité cosinus pour créer un système de recommandation. Cette approche nous permettra d'exploiter efficacement les caractéristiques et les préférences des clients existants afin de proposer des recommandations personnalisées et pertinentes.

De plus, pour accueillir les nouveaux clients, nous avons prévu de mettre en place un deuxième

système basé sur Neo4j, une base de données graphes hautement performante. Grâce à Neo4j, nous pourrons modéliser les relations entre les clients, les partenaires et d'autres entités, ce qui facilitera la découverte de nouveaux éléments intéressants pour les nouveaux arrivants. Cette approche novatrice nous permettra d'offrir une expérience utilisateur exceptionnelle dès le premier jour d'utilisation de notre application.

1.3 Méthodologie de gestion du projet

Pour assurer une gestion efficace du projet, une bonne organisation est nécessaire. Nous avons choisi d'adopter la méthode agile SCRUM pour la conception et le développement de notre solution.

1.3.1 Méthode agile

La méthode agile est une approche de gestion de projet qui favorise la flexibilité, la collaboration et l'adaptation aux changements tout au long du processus de développement. Elle se caractérise par des cycles itératifs et un fort engagement des parties prenantes.

1.3.2 SCRUM

Scrum est un cadre de gestion de projet agile qui favorise la collaboration, l'adaptabilité et la livraison itérative de produits. Il se caractérise par des cycles de travail appelés "sprints" et une approche centrée sur l'équipe, l'auto-organisation et l'amélioration continue.

1.3.3 Méthodes Agiles : SCRUM (Rôles)

SCRUM s'articule autour de trois rôles particuliers, chacun ayant une fonction bien précise :

- **Scrum Master** : est le facilitateur agile qui guide l'équipe de développement dans l'application de Scrum et favorise la collaboration et l'amélioration continue.
- **Product Owner** : est le responsable du produit, définissant la vision et les priorités, et collaborant avec l'équipe de développement pour assurer sa création et son amélioration continue.
- **L'équipe de développement** : est une équipe auto-organisée et pluridisciplinaire, responsable de la réalisation des éléments du backlog du produit lors des sprints.

Le tableau 1.2 présente les acteurs du projet conformément à la méthodologie Scrum :

Rôle	Acteur
Scrum Master	Mariem Knani
Product Owner	Islem Dahmen
Équipe	<ul style="list-style-type: none"> — Maha Kchaich — Abir Dahmen — Achref Taleb

TABLE 1.2 – Acteurs du projet selon "Scrum"

1.3.4 Méthodes Agiles : SCRUM (Pratiques)

Les artefacts Scrum sont des éléments permettant de faire fonctionner le cadre de travail Scrum :

- **Le Product Backlog** : est une liste ordonnée de toutes les fonctionnalités, les exigences et les améliorations souhaitées pour le produit.
- **Le Sprint Backlog** : est une liste des éléments du Product Backlog qui ont été sélectionnés pour être réalisés pendant un Sprint spécifique.
- **L'Increment ou Le sprint** : est une itération de développement, aboutissant à un incrément de fonctionnalité potentiellement livrables.
- **Les User Stories** : sont des descriptions courtes des fonctionnalités du produit, du point de vue de l'utilisateur.
- **Les Reunions Scrum** : sont des réunions régulières et structurées qui font partie du cadre Scrum. Ces réunions permettent à l'équipe de se synchroniser, de planifier et d'adapter son travail.

La Figure 1.4 représente le nombre de sprints et la répartition des tâches dans chaque sprint de manière visuelle.

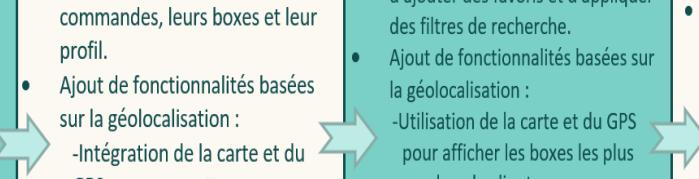
Sprint1: Partie Web	Sprint2: Partie mobile côté partenaire	Sprint3: Partie mobile côté client	Sprint4: Module AI
<ul style="list-style-type: none"> • Mise en place de la structure du tableau de bord administrateur. • Création des pages pour gérer les partenaires, les clients, les boxes et les commandes. • Implémentation des fonctionnalités de base pour ajouter, modifier et supprimer des partenaires, des clients, des boxes et des commandes. • Ajout de fonctionnalités avancées pour la gestion des partenaires, des clients, des commandes et des boxes (par exemple, filtres, recherche, pagination). 	<ul style="list-style-type: none"> • Développement de l'application mobile pour les partenaires. • Développement des fonctionnalités permettant aux partenaires de gérer leurs commandes, leurs boxes et leur profil. • Ajout de fonctionnalités basées sur la géolocalisation : <ul style="list-style-type: none"> -Intégration de la carte et du GPS pour permettre aux partenaires de gérer leur adresse. -Affichage de l'emplacement actuel du partenaire sur la carte. • Intégration d'un système de génération de code QR pour confirmer la récupération des boxes. 	<ul style="list-style-type: none"> • Développement de l'application mobile pour les clients. • Implémentation des fonctionnalités permettant aux clients de passer des commandes, de consulter les boxes, d'ajouter des favoris et d'appliquer des filtres de recherche. • Ajout de fonctionnalités basées sur la géolocalisation : <ul style="list-style-type: none"> -Utilisation de la carte et du GPS pour afficher les boxes les plus proches du client. -Intégration de l'emplacement actuel du client. -Possibilité pour le client de modifier la distance de recherche sur la carte pour afficher les partenaires situés dans un rayon spécifique. • Intégration du système de génération de code QR pour les clients lorsqu'ils récupèrent leur box. 	<ul style="list-style-type: none"> • Implémentation des fonctionnalités des recommandations de boxes en utilisant l'IA. • Implémentation des fonctionnalités des recommandations basé sur Neo4j.

FIGURE 1.4 – Répartition de Sprints

1.4 Planification du projet

Le diagramme de gantt est un outil graphique qui représente la gestion du projet dans le temps ce qui facilite sa réalisation. En effet, la figure 1.5 représente l'avancement des activités et des tâches exécutées tout le long de notre projet.

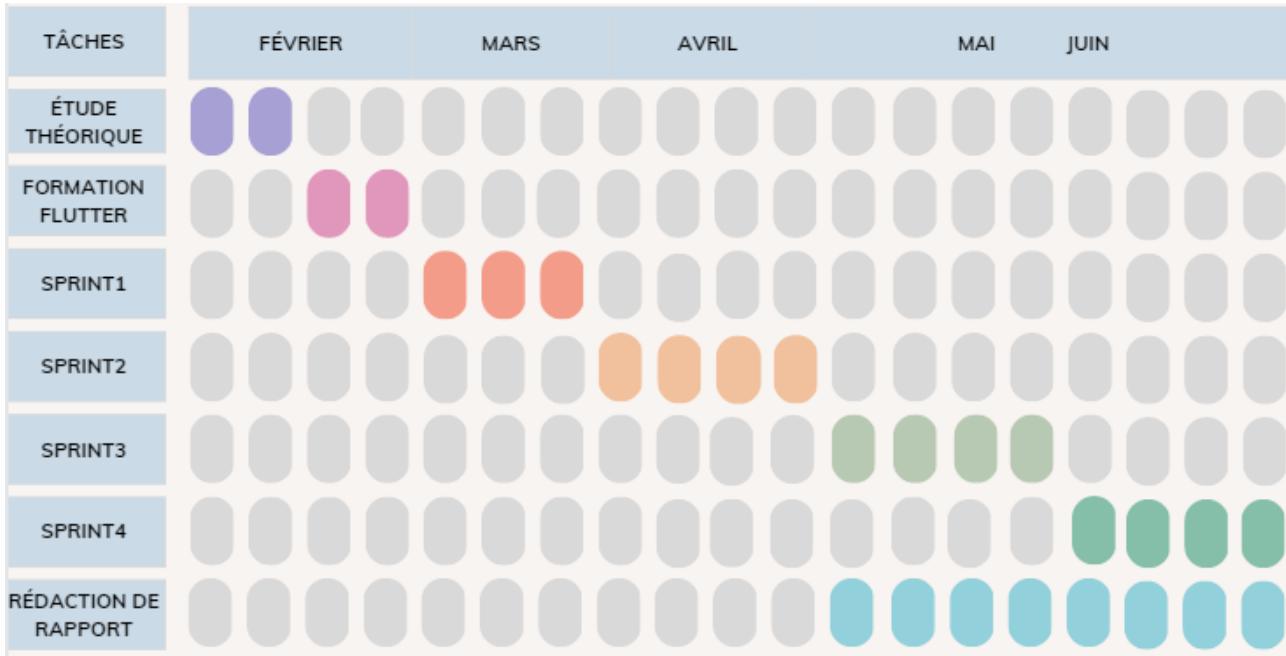


FIGURE 1.5 – Diagramme de GANTT

Conclusion

Au cours de ce chapitre, nous avons entamé en présentant le contexte général du projet. Tout d'abord, nous avons introduit l'organisme d'accueil. Ensuite, nous avons abordé la problématique. En troisième lieu, nous avons réalisé une analyse des applications existantes. Par la suite, nous avons décrit la solution proposée. Enfin, nous avons exposé la méthodologie de gestion du projet et établi la planification. Dans le prochain chapitre, nous procéderons à une analyse approfondie et à la spécification des besoins.

Chapitre 2

Analyse et spécification des besoins

Introduction

Dans ce deuxième chapitre, nous allons nous focaliser sur l'étape d'analyse et de spécification des besoins qui va faciliter le développement de notre projet. Nous allons commencer par identifier les acteurs. Ensuite, nous allons spécifier les exigences fonctionnelles et non fonctionnelles auxquelles la solution proposée devra répondre. Enfin, nous allons formaliser les besoins à travers les diagrammes de cas d'utilisation et les diagrammes de séquence système du langage UML. Ces diagrammes expliquent les principales fonctionnalités de notre application.

2.1 Identification des acteurs

Notre solution comprendra deux parties : une partie web et une partie mobile. Les acteurs sont les entités externes qui interagissent avec le système. Dans notre cas, les acteurs primaires sont l'administrateur, le partenaire et le client. Nous avons également des acteurs secondaires, tels que le système de génération des codes QR, le système d'archivage des boxes et le système de recommandation des boxes. En identifiant ces acteurs, nous serons en mesure de mieux comprendre leurs besoins et de concevoir des fonctionnalités spécifiques pour chaque partie impliquée.

La partie

2.1.1 Partie Web (Back-office)

Elle sera utilisée par :

- **Administrateur** : Cet acteur joue le rôle de responsable système. Il peut réaliser la majorité des opérations de gestion, comme la gestion des clients, la gestion des partenaires, la gestion des boxes, l'analyse des données et le suivi des réservations.

2.1.2 Partie Mobile

Elle sera utilisée par :

1. **Les acteurs primaires** :

- **Partenaire** : Son rôle est de vendre les aliments invendus à la fin de la journée à des prix réduits. Le partenaire peut publier des boxes sur l'application, gérer les boxes et suivre les ventes. Son objectif principal est de minimiser le gaspillage alimentaire tout en augmentant ses revenus en vendant des produits qui seraient autrement perdus.
- **Client** : Son rôle dans l'application est de réserver des boxes qui contiennent les aliments. Le client peut chercher des magasins à proximité, voir les boxes disponibles à l'achat et les réserver. Son objectif principal est de minimiser le gaspillage alimentaire tout en réalisant des économies sur ses achats alimentaires quotidiens.

2. Les acteurs secondaires :

- **Le système de génération de QR code** : Nous utilisons ce système dans notre système de gestion des commandes pour les clients et les partenaires. Ce système attribue un QR code unique à chaque commande, que le client scanne pour indiquer qu'il a reçu sa commande.
- **Le système de recommandation des boxes** : Nous concevons ce système afin de proposer de manière intelligente des recommandations de box adaptées à chaque client en fonction de leurs besoins et préférences.
- **Le système d'archivage des boxes** : Nous utilisons ce système dans la gestion des boxes pour stocker les boxes qui sa fonction se termine dans l'application, afin de faciliter les tâches pour les partenaires et améliorer la lisibilité.

2.2 Spécification des besoins

Le but de cette section consiste à rassembler, étudier et déterminer les besoins essentiels ainsi que les caractéristiques clés du projet. L'accent est mis sur les exigences fonctionnelles et non fonctionnelles qui sont nécessaires pour proposer des services conviviaux et faciles à utiliser.

2.2.1 Les besoins fonctionnels

Un ensemble de besoins fonctionnels est mis en œuvre par notre système afin de répondre aux attentes des utilisateurs, nous allons classer les besoins fonctionnels du système selon nos différents acteurs

- **Administrateur :**
 - Gérer son profil : modifier ses informations personnelles, telles que son nom, son adresse e-mail ou ses informations de contact, ainsi q'il peut faire la mise à jour de son mot de passe.
 - Consulter le tableau de bord : cette fonctionnalité fournit une vue d'ensemble des statistiques de l'application, permettant à l'administrateur de comprendre rapidement comment l'application est utilisée.

- Gérer les partenaires : il accepte les partenaires et accomplit diverses tâches liées à leur gestion, notamment l'ajout, la modification ou la suppression, ainsi que la recherche et le filtrage selon certains critères. De plus, il a la capacité d'activer ou de désactiver un partenaire.
- Gérer les clients : il peut effectuer plusieurs tâches liées à la gestion des clients, telles que l'ajout, la modification ou la suppression. Il peut également chercher et filtrer les clients selon certains critères. Il a la possibilité d'activer ou désactiver un client.
- Gérer les boxes : une fois que l'administrateur a accepté un box, il est capable d'effectuer diverses tâches relatives à la gestion de celui-ci, telles que l'ajout, la modification ou la suppression, ainsi que la recherche et le filtrage. En outre, il dispose de la capacité de modifier son état.
- Gérer les commandes : il est capable d'accomplir diverses tâches relatives à la gestion des commandes, notamment l'ajout, la modification de leur statut ou la suppression. De plus, il a la capacité de rechercher et de filtrer les commandes en fonction de certains critères.

— **Partenaire :**

- Gérer son profil : implique la prise en charge ses informations personnelles telles que le nom, l'adresse e-mail, le numéro de téléphone, les coordonnées, etc., en permettant la modification de ces données. De plus, il est possible de mettre à jour la photo de profil et de changer le mot de passe. En outre, il est également possible de consulter les statistiques relatives aux revenus.
- Gérer les boxes : assurer la gestion des opérations d'ajout et de consultation en fonction de leur état, avec la possibilité de les modifier **si ils se trouvent à l'état "en attente".**
s'ils sont "en attente"
- Consulter les commandes : gérer les opérations de vérification des commandes et les consulter selon leur état.

— **Client :**

- Gérer son profil : il peut prendre en charge la gestion de son profil en modifiant ses informations personnelles, telles que son nom, son adresse e-mail, etc. De plus, il a la possibilité de mettre à jour son mot de passe et de consulter certaines statistiques.
- Réserver box : à travers cette fonctionnalité, l'utilisateur a la possibilité de réserver un box en passant une commande pour celui qu'il a choisi.
- Valider commande : une fois que la commande est prête, le client peut se rendre sur place pour récupérer son box réservé, valider sa commande et donner une évaluation de l'expérience d'achat.
- Gérer les favoris : il a la possibilité d'ajouter des boxes et des partenaires à sa liste de favoris, leur permettant ainsi d'être facilement accessibles et consultables. Il peut aussi les supprimer de la liste des favoris.

- Consulter boxes : il a la possibilité de visualiser les boxes disponibles, ainsi que les boxes recommandés spécifiquement pour lui. Il peut également accéder aux détails de chaque box, y compris les informations sur le partenaire qui le propose.
- Sélectionner boxes : il a la possibilité de sélectionner des boxes en les filtrant selon ses préférences et certains critères. Il peut effectuer une recherche de boxes ou choisir une distance spécifique sur une carte graphique afin de visualiser les partenaires à proximité de sa position.

2.2.2 Les besoins non fonctionnels

cités

Outre que les besoins fonctionnels cité précédemment, pour bien répondre aux exigences des utilisateurs, notre système devra pouvoir assurer les besoins non fonctionnels suivants :

- **Exigences de qualité :**
 1. Notre application doit offrir une interface conviviale et ergonomique exploitable par l'utilisateur en envisageant toutes les interactions possibles à l'écran du support tenu.
 2. L'application bénéficie des interfaces adaptées élaborées par l'équipe de créateurs de la société Webify.
 3. Il est nécessaire que la plateforme soit capable de prendre en charge plusieurs langues, notamment l'anglais et le français doivent être intégrés.
- **Exigences de portabilité :**
 1. Utilisation de technologies multiplateformes, l'application doit être compatible avec les différents systèmes d'exploitation et appareils mobiles pour une utilisation optimale.
 2. Angular et Flutter sont tous deux des frameworks de développement d'applications multiplateformes, ce qui signifie qu'ils vous permettent de créer des applications pour plusieurs plateformes (comme le web, iOS et Android) en utilisant un code source commun. Ils offrent tous deux certains avantages en termes de portabilité, c'est-à-dire la facilité de transfert d'une plateforme à une autre.
- **Exigences de performance :**
 1. Il est essentiel d'optimiser le temps de réponse et de chargement des données afin de garantir une expérience utilisateur fluide et efficace. En outre, il est primordial que le système fonctionne sans erreurs, assurant ainsi un fonctionnement fiable et sans problème pour les utilisateurs. Il est également important de souligner que le délai de réponse ne doit pas dépasser 60 secondes afin d'éviter toute frustration de l'utilisateur et de garantir une utilisation efficace et ininterrompue de l'application.
 2. Le système doit être capable de gérer simultanément et sans interruption la manipulation d'une grande quantité de données.
 3. Toutes les requêtes sont pertinentes à 99 pour cent.
 4. Le système supporte plus de 10000 comptes.
- **Contraintes de conception :**

1. Enregistrement des fichiers : Les fichiers utilisés dans notre système, tels que les images, ils seront enregistrés sur une source externe et référencés dans la base de données avec leurs URLs.

2.3 Modélisation des besoins

Pour obtenir une vue globale des exigences de notre application, nous allons présenter une modélisation des besoins fonctionnels en ayant recours aux notions du langage "UML".

2.3.1 Langage de modélisation UML

Le langage UML (Unified Modeling Language) est constitué de diagrammes intégrés utilisés par les développeurs informatiques pour la représentation visuelle des objets, des états et des processus dans un logiciel ou un système. Le langage de modélisation peut servir de modèle pour un projet et garantir une architecture d'information structurée, il peut également aider les développeurs à présenter leur description d'un système d'une manière compréhensible pour les spécialistes externes.^[9] Pour cela, nous avons utilisé "UML" pour la modélisation.

2.3.2 Diagramme de cas d'utilisation Global de la partie Web (Back-office)

La figure 2.1 suivante présente le diagramme de cas d'utilisation général de la partie web qui permet d'identifier les grandes fonctionnalités du futur système et les possibilités d'interaction entre notre système et l'administrateur.

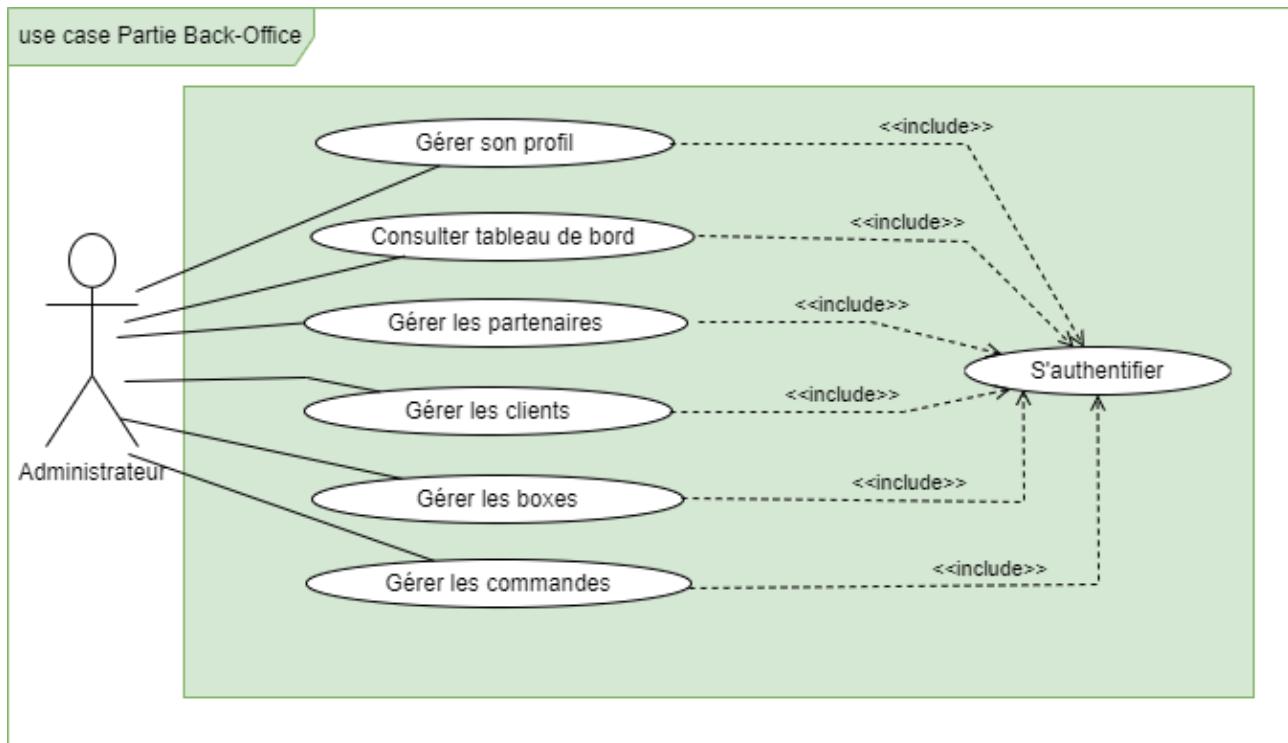


FIGURE 2.1 – Diagramme de cas d'utilisation générale du partie Web

2.3.3 Diagramme de cas d'utilisation Global de la partie mobile

La figure 2.2 suivante présente le diagramme de cas d'utilisation global de la partie mobile illustre les principales fonctionnalités du notre système.

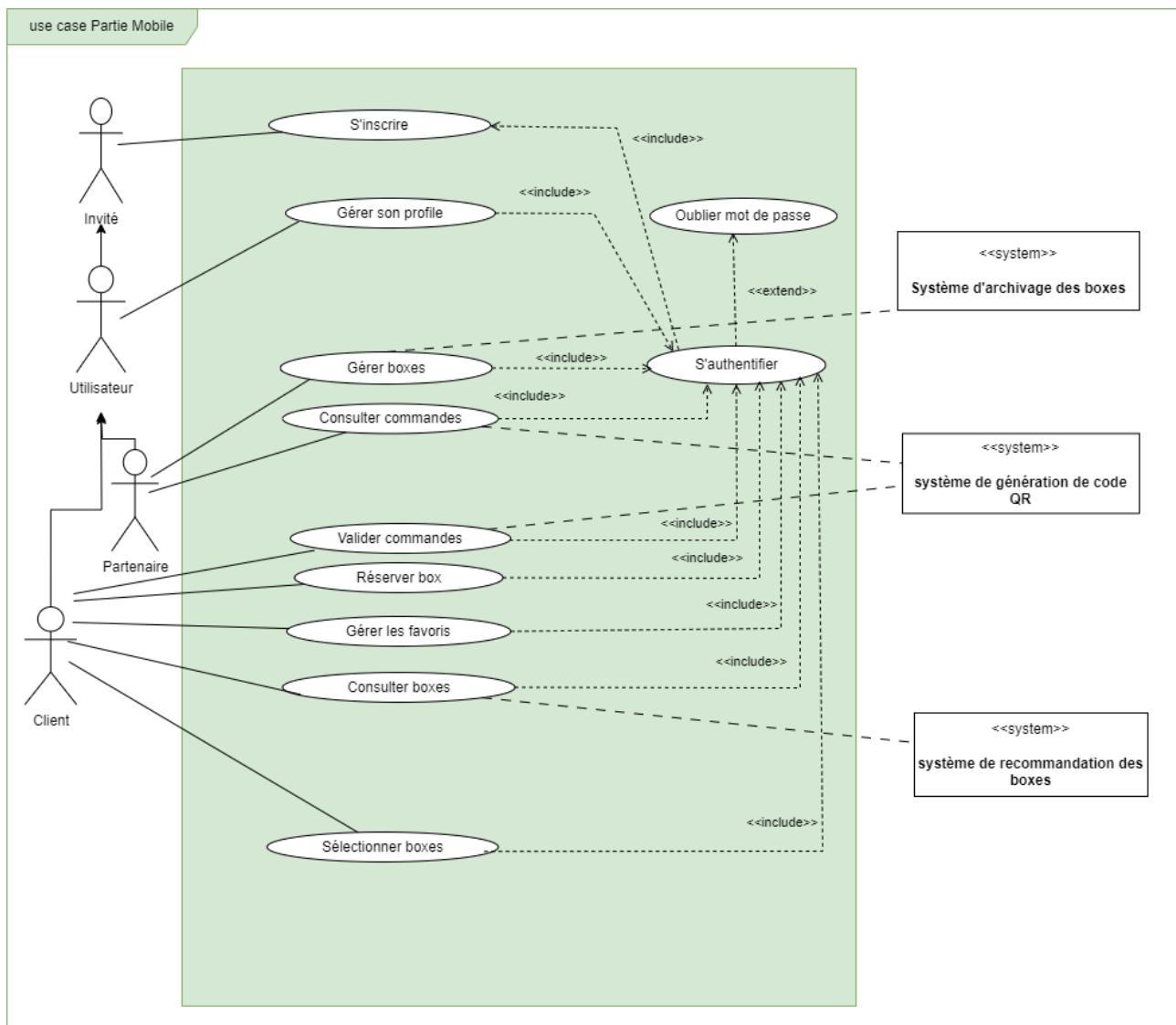


FIGURE 2.2 – Diagramme de cas d'utilisation générale du partie mobile général de la partie

2.4 Modélisation fonctionnelle des fonctionnalités système

Dans cette partie, nous détaillons les cas d'utilisation les plus importants offerts par notre système .

2.4.1 Sprint 1 : Partie Web

2.4.1.1. Backlog du sprint 1

Le tableau 2.1 décrit le Backlog du produit de notre partie web :

Numéro	User Story	Acteur	Tâches	Durée	Priorité
1	Authentification	Administrateur	Connexion Administrateur	2	1
2	Gérer profil	Administrateur	Consulter et modifier son profil	1	2
3	Gérer les partenaires	Administrateur	<ul style="list-style-type: none"> — Ajouter partenaire. — Modifier partenaire. — Supprimer partenaire. — Chercher partenaire. — Filtrer partenaire. — Afficher détails partenaire. 	5	1
4	Gérer les clients	Administrateur	<ul style="list-style-type: none"> — Ajouter client. — Modifier client. — Supprimer client. — Chercher client. — Filtrer client. — Afficher détails client. 	5	1
5	Gérer les boxes	Administrateur	<ul style="list-style-type: none"> — Ajouter box. — Modifier box. — Supprimer box. — Chercher box. — Filtrer box. — Afficher détails box. 	5	1
6	Gérer les commandes	Administrateur	<ul style="list-style-type: none"> — Ajouter commande. — Modifier commande. — Supprimer commande. — Chercher commande. — Filtrer commande. — Afficher détails commande. 	5	1

TABLE 2.1 – Backlog du produit de partie web

2.4.1.2. Diagramme de navigation système (DNS) de sprint 1 :

La figure 2.3 représente les différents pages qui composent la partie web, les transitions entre ces écrans ainsi que les informations sur les actions disponibles à partir de chaque écran.

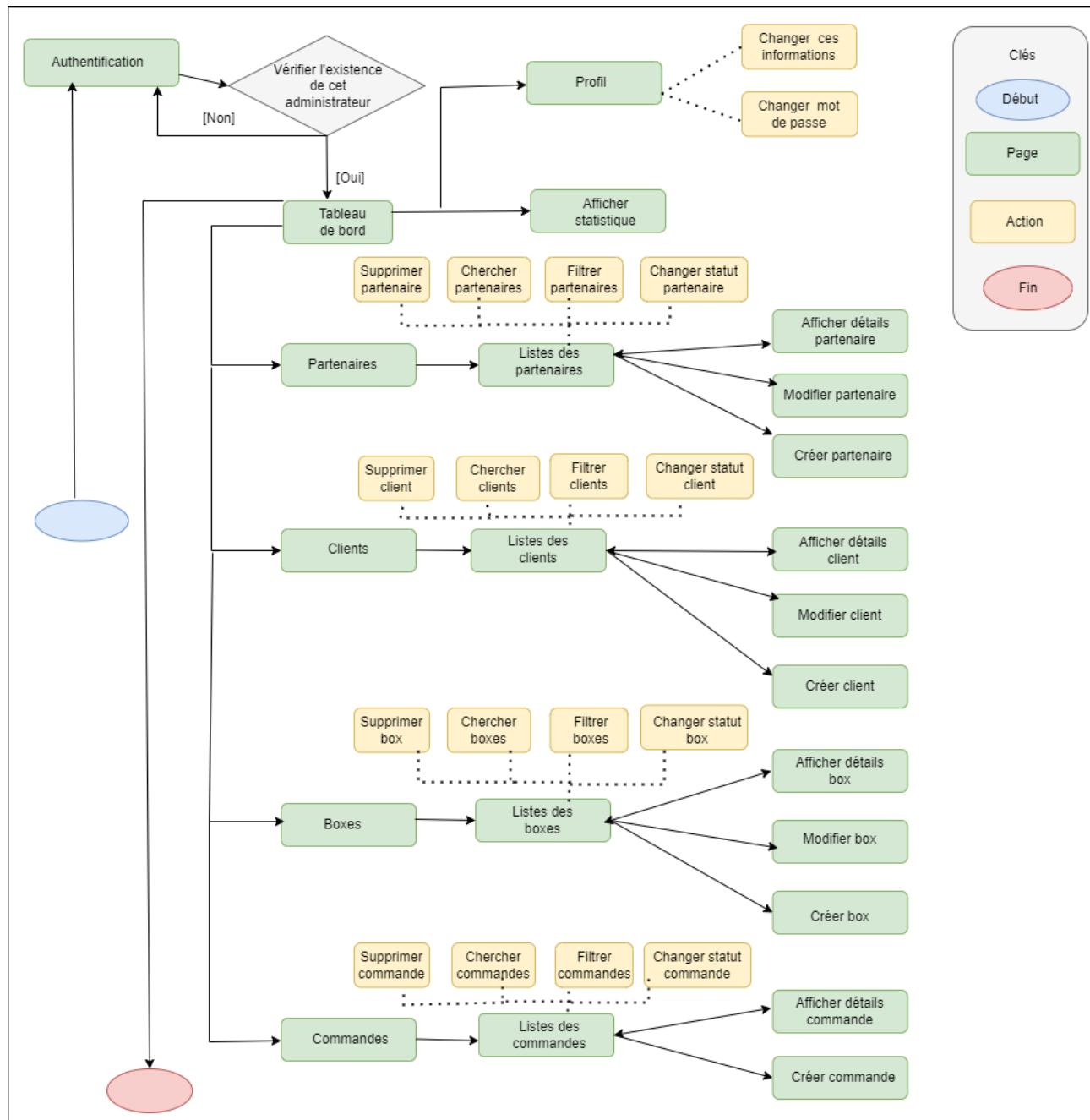


FIGURE 2.3 – Diagramme de navigation de partie Web

2.4.1.3. Raffinement du cas d'utilisation « Gérer les partenaires » :

La figure 2.4 présente le raffinement **du** cas d'utilisation gérer les partenaires, suivie par leur description textuelle.

du

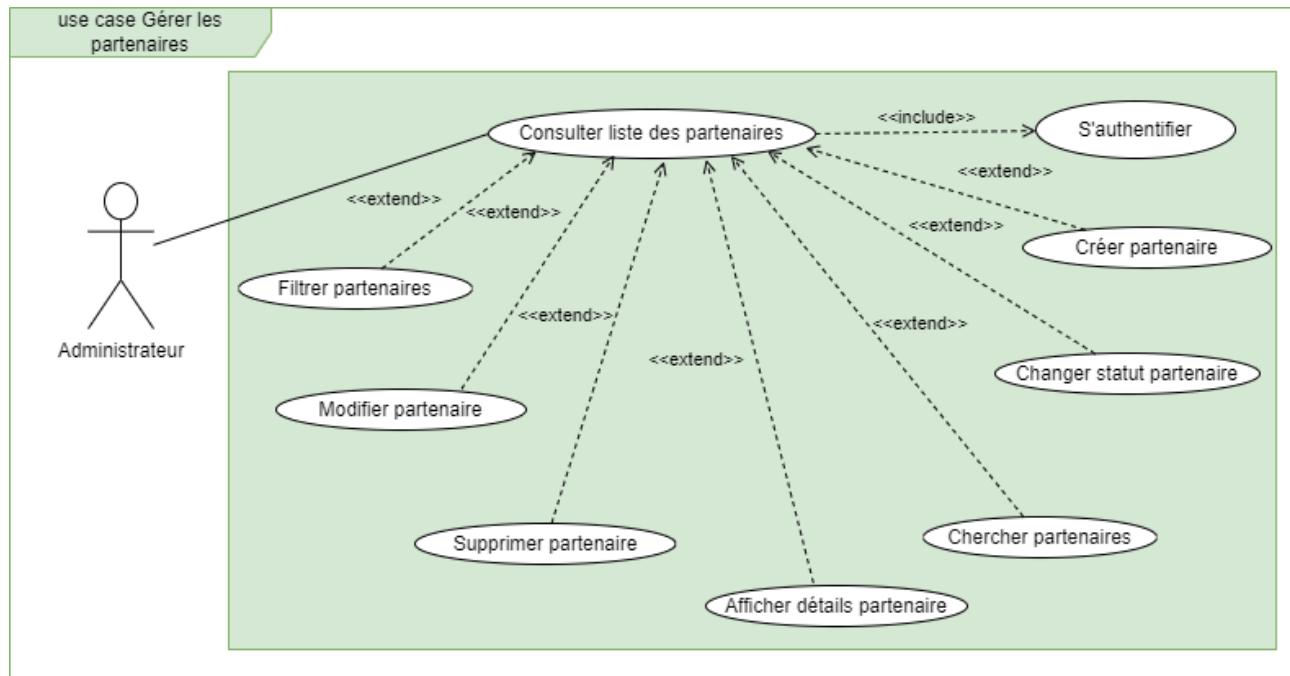


FIGURE 2.4 – Diagramme détaillé de cas d'utilisation « Gérer les partenaires »

La description de scénarios des cas d'utilisation : « Gérer les partenaires » est donnée par le tableau 2.2 :

Titre	Gérer les partenaires
Acteurs	Administrateur.
Objectif	L'administrateur peut gérer les partenaires en accédant à une liste complète des partenaires, où il peut consulter les détails de chacun, créer de nouveaux partenaires, les modifier, les supprimer, les filtrer par catégorie ou effectuer des recherches parmi les partenaires.
Précondition	Administrateur authentifié.
Post-condition	Mise à jour sur un partenaire ou recherche des partenaires.

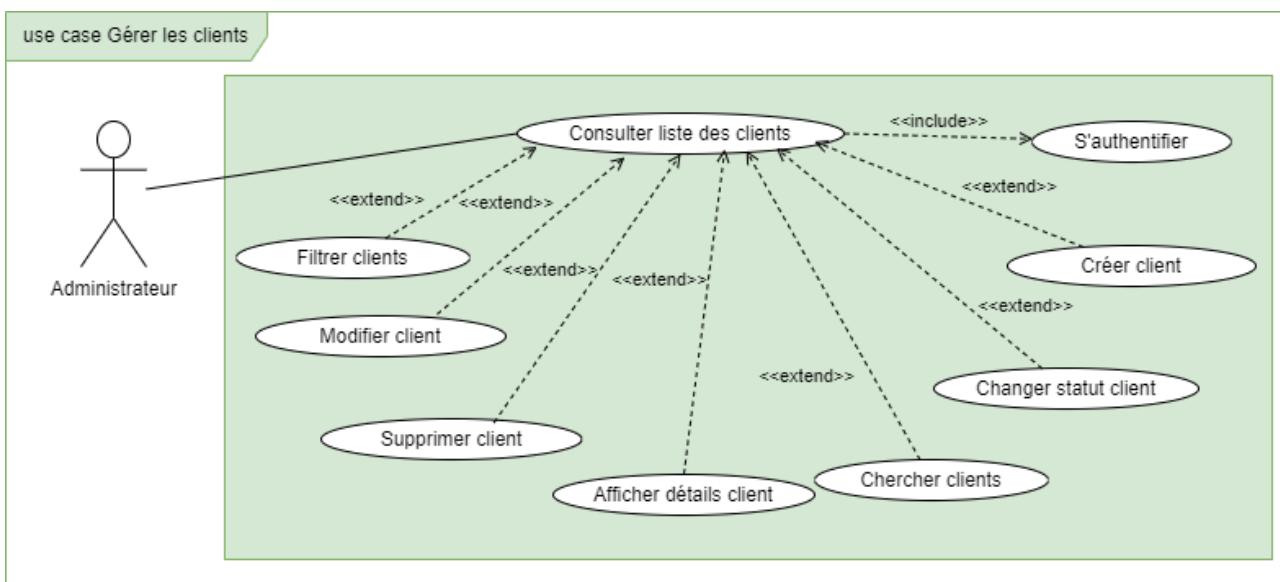
Scénario nominal	<p>[S1] : Mise à jour sur un partenaire :</p> <ul style="list-style-type: none"> (a) L'administrateur fait son choix sur "Partners". (b) Le système affiche la liste des partenaires. [A1] (c) Lorsque l'administrateur souhaite modifier le partenaire choisi, il procède en cliquant sur le bouton de modification pour le sélectionner. (d) Le système affiche un formulaire rempli avec les informations de ce partenaire et deux autres formulaires pour le changement d'image et le mot de passe. (e) L'administrateur apporte des modifications aux informations de ce partenaire et les valide. [A2] (f) Le système vérifie les champs, modifie le partenaire et affiche un message de succès puis redirige l'administrateur vers la liste des partenaires. <p>[S2] : Recherche des partenaires :</p> <ul style="list-style-type: none"> (a) L'administrateur fait son choix sur "Partners". (b) Le système affiche la liste des partenaires. [A1] (c) L'administrateur recherche des partenaires en cliquant sur la barre de recherche et en saisissant soit leur adresse email , soit leur numéro de téléphone. (d) Une fois cette étape franchie, une liste de partenaires correspondant aux critères saisis par l'administrateur s'affiche. [A2]
Scénario alternatif	<ul style="list-style-type: none"> — [S1.A1] / [S2.A1] : la liste des partenaires est vide. — [S1.A2.a] : Le mot de passe ou le numéro de téléphone saisi n'est pas valide : la longueur du mot de passe est inférieure à 6 et la longueur du numéro de téléphone est supérieure à 8. Dans de tels cas, le système envoie un message d'erreur expliquant les conditions de validation pour ce champ. — [S1.A2.b] : L'email est déjà utilisé : le système envoie une alerte. — [S1.A2.c] : Vérification des horaires : si l'heure de fermeture est antérieure à l'heure d'ouverture, un message d'erreur s'affiche. — [S2.A2.a] : il n'y a pas de partenaires correspondant aux critères saisis par l'administrateur, liste vide de partenaires s'affiche.

Exigences supplémentaires	<p>[S1] : Mise à jour sur un partenaire : le bouton "Save" de validation devient désactivé si :</p> <ul style="list-style-type: none"> — L'un des champs est vide. — La longueur du nom est inférieure à 3. — La longueur du numéro de téléphone est inférieure à 8. De plus, dans le champ de téléphone, seuls les chiffres sont autorisés. — Le format de l'e-mail n'est pas valide. <p>[S2] : Recherche des partenaires :</p> <ul style="list-style-type: none"> — L'administrateur peut effectuer à la fois le filtrage et la recherche des partenaires au même temps.
---------------------------	---

TABLE 2.2 – Cas d'utilisation « Gérer les partenaires »

2.4.1.4. Raffinement du cas d'utilisation « Gérer les clients » :

La figure 2.5 présente le raffinement **de** cas d'utilisation gérer les clients, suivie par leur description textuelle.

**FIGURE 2.5 – Diagramme détaillé de cas d'utilisation « Gérer les clients »**

La description de scénarios des cas d'utilisation : « Gérer les clients » est donnée par le tableau 2.3 :

Titre	Gérer les clients
Acteurs	Administrateur.

Objectif	En accédant à une liste exhaustive des clients, l'administrateur dispose de diverses possibilités de gestion. Il peut consulter les informations détaillées de chaque client, créer de nouveaux clients, les modifier ou les supprimer. De plus, il a la possibilité de filtrer les clients par catégorie ou d'effectuer des recherches selon certains critères spécifiques.
Précondition	Administrateur authentifié.
Post-condition	Création d'un client avec succès.
Scénario nominal	<ul style="list-style-type: none"> (a) L'administrateur fait son choix sur "Users". (b) Le système affiche la liste des clients. (c) L'administrateur clique sur le bouton 'Add' pour ajouter un nouveau client. (d) Le système affiche le formulaire à remplir. (e) L'administrateur remplit le formulaire puis clique sur le bouton "Save". (f) Le système vérifie la saisie, enregistre un nouveau client, affiche un message de succès puis redirige l'administrateur vers la liste des clients. [A1]
Scénario alternatif	<ul style="list-style-type: none"> — [A1.a] : L'email est déjà utilisé : le système envoie un message d'erreur. — [A1.b] : Le numéro de téléphone saisi n'est pas valide : la longueur du numéro de téléphone est supérieure à 8. Dans ce cas, le système envoie un message d'erreur expliquant les conditions de validation pour ce champ.
Exigences supplémentaires	<p>le bouton "Save" de validation devient désactivé si :</p> <ul style="list-style-type: none"> — L'un des champs est vide. — La longueur du nom est inférieure à 3. — La longueur du numéro de téléphone est inférieure à 8. De plus, dans le champ de téléphone, seuls les chiffres sont autorisés. — Le format de l'e-mail n'est pas valide. — La longueur du mot de passe est inférieure à 6. <p>Le mot de passe ne doit pas être enregistré en clair dans la base de données. Il doit être haché.</p>

TABLE 2.3 – Description textuelle du cas « Gérer les clients »

2.4.1.5. Raffinement du cas d'utilisation « Gérer les boxes » :

La figure 2.6 présente le raffinement de cas d'utilisation gérer les boxes, suivie par leur description textuelle.

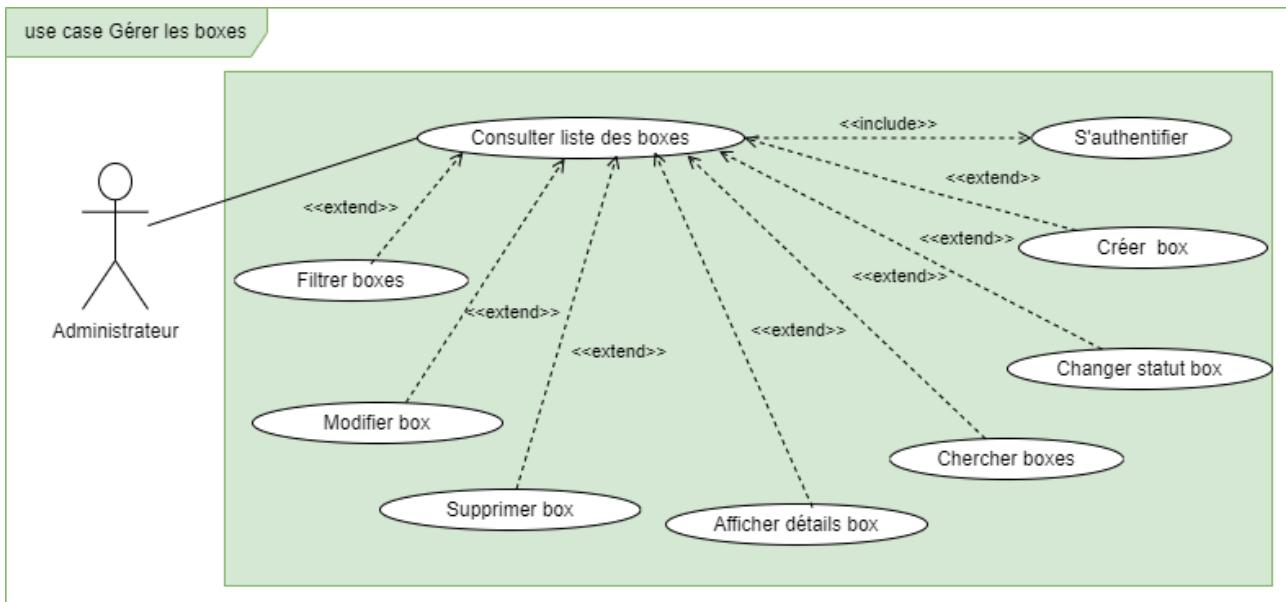


FIGURE 2.6 – Diagramme détaillé de cas d'utilisation « Gérer les boxes »

La description de scénarios des cas d'utilisation : « Gérer les boxes » est donnée par le tableau 2.4 :

Titre	Gérer les boxes
Acteurs	Administrateur.
Objectif	L'administrateur a un large éventail d'options de gestion en utilisant la liste des boxes. Il peut examiner les informations détaillées de chaque box, créer de nouveaux boxes, les modifier ou les supprimer. En outre, il peut filtrer les boxes en fonction de leur état et effectuer des recherches en se basant sur des critères spécifiques.
Précondition	Administrateur authentifié.
Post-condition	Suppression d'un box ou bien filtrage des boxes.

Scénario nominal	<p>[S1] : Supprimer box :</p> <ul style="list-style-type: none"> (a) L'administrateur fait son choix sur "Boxes". (b) Le système affiche la liste des boxes. [A1] (c) Lorsque l'administrateur souhaite supprimer une box choisie, il procède en cliquant sur l'icône de suppression pour la sélectionner. (d) Le système affiche une fenêtre contextuelle pour confirmer la suppression ou l'annuler. (e) L'administrateur confirme la suppression. (f) Le système affiche un message de succès et rafraîchit la liste des boxes affichée. <p>[S2] : Filtrer les boxes :</p> <ul style="list-style-type: none"> (a) L'administrateur fait son choix sur "Boxes". (b) Le système affiche la liste des boxes. [A1] (c) L'administrateur accède à la liste de filtrage à partir de laquelle il peut sélectionner l'un des états de box. (d) Le système va afficher la liste des boxes qui se trouvent dans cet état. [A2]
Scénario alternatif	<ul style="list-style-type: none"> — [S1.A1] [S2.A1] : La liste de boxes est vide. — [S2.A2] : Le système affiche "No Data Found", c'est-à-dire qu'il n'existe pas de boxes dans l'état sélectionné.
Exigences supplémentaires	<ul style="list-style-type: none"> — L'interface doit être ergonomique. — Les messages doivent être compréhensibles et clairs. — L'administrateur peut effectuer à la fois le filtrage et la recherche des partenaires au même temps.

TABLE 2.4 – Description textuelle du cas «Gérer les boxes»

2.4.2 Sprint 2 : Partie mobile côté partenaire

2.4.2.1. Backlog du sprint 2

Le tableau 2.5 décrit le Backlog du produit de notre partie mobile côté partenaire :

Numéro	User Story	Acteur	Tâches	Durée	Priorité
1	Authentification et inscription	Partenaire	<ul style="list-style-type: none"> — Inscrire partenaire — Connexion partenaire 	7	1

2	Gérer profil	partenaire	<ul style="list-style-type: none"> — Consulter et modifier son profil — Consulter ses statistiques — Gérer son adresse 	7	2
3	Gérer ses commandes	Partenaire	<ul style="list-style-type: none"> — consulter ses commandes. — valider ses commandes. 	7	1
4	Gérer les boxes	Partenaire	<ul style="list-style-type: none"> — Ajouter box. — consulter ses boxes. — modifier box. 	7	1

TABLE 2.5 – Backlog du produit de partie mobile côté partenaire

2.4.2.2. Diagramme de navigation système (DNS) de sprint 2 :

La figure 2.7 représente les différents pages qui composent la partie mobile côté partenaire, les transitions entre ces écrans ainsi que les informations sur les actions disponibles à partir de chaque écran.

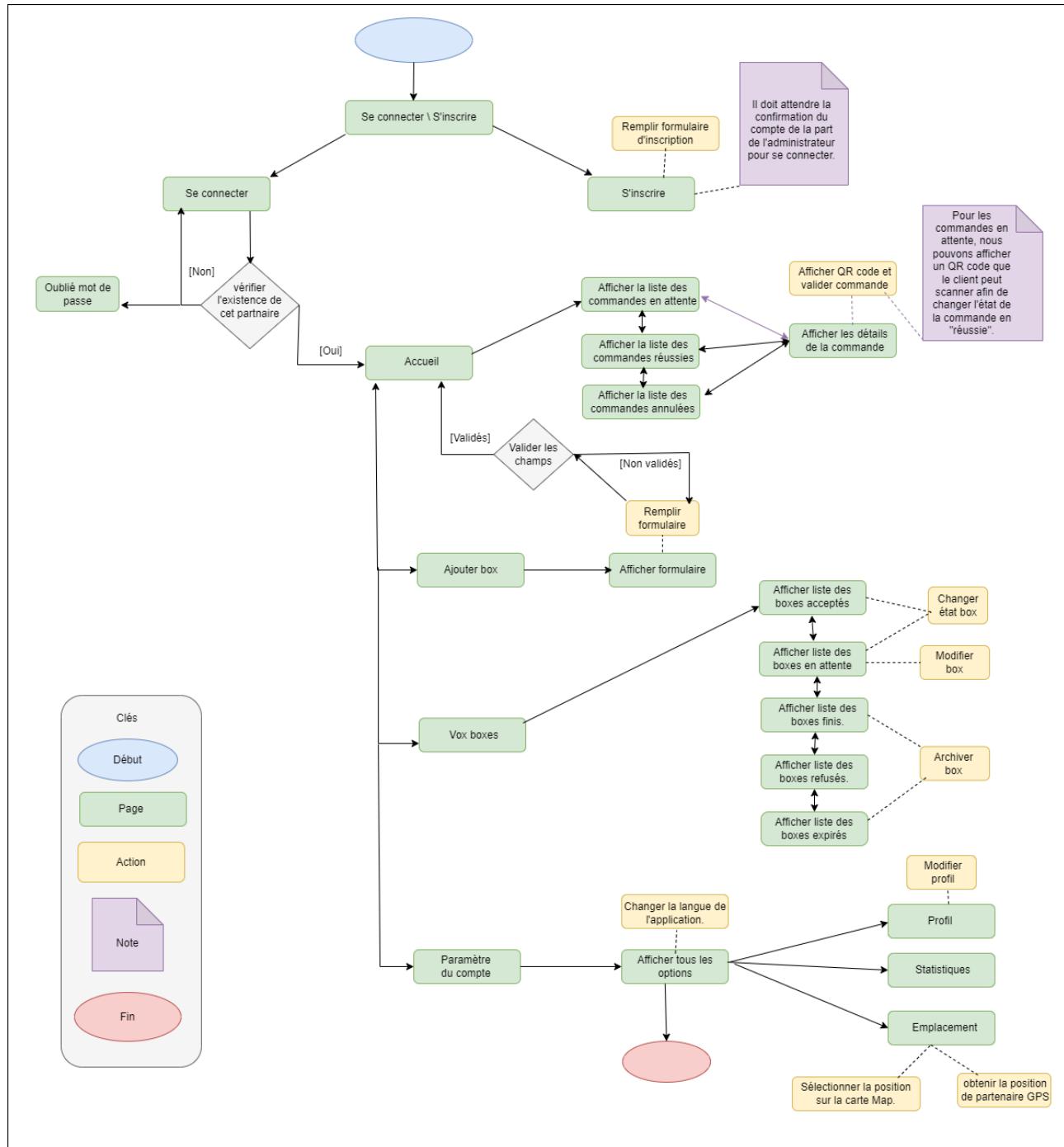


FIGURE 2.7 – Diagramme de navigation de partie mobile côté partenaire

2.4.2.3. Diagramme de séquence système « S'authentifier » :

La figure 2.9 présente le diagramme de séquence système s'authentifier.

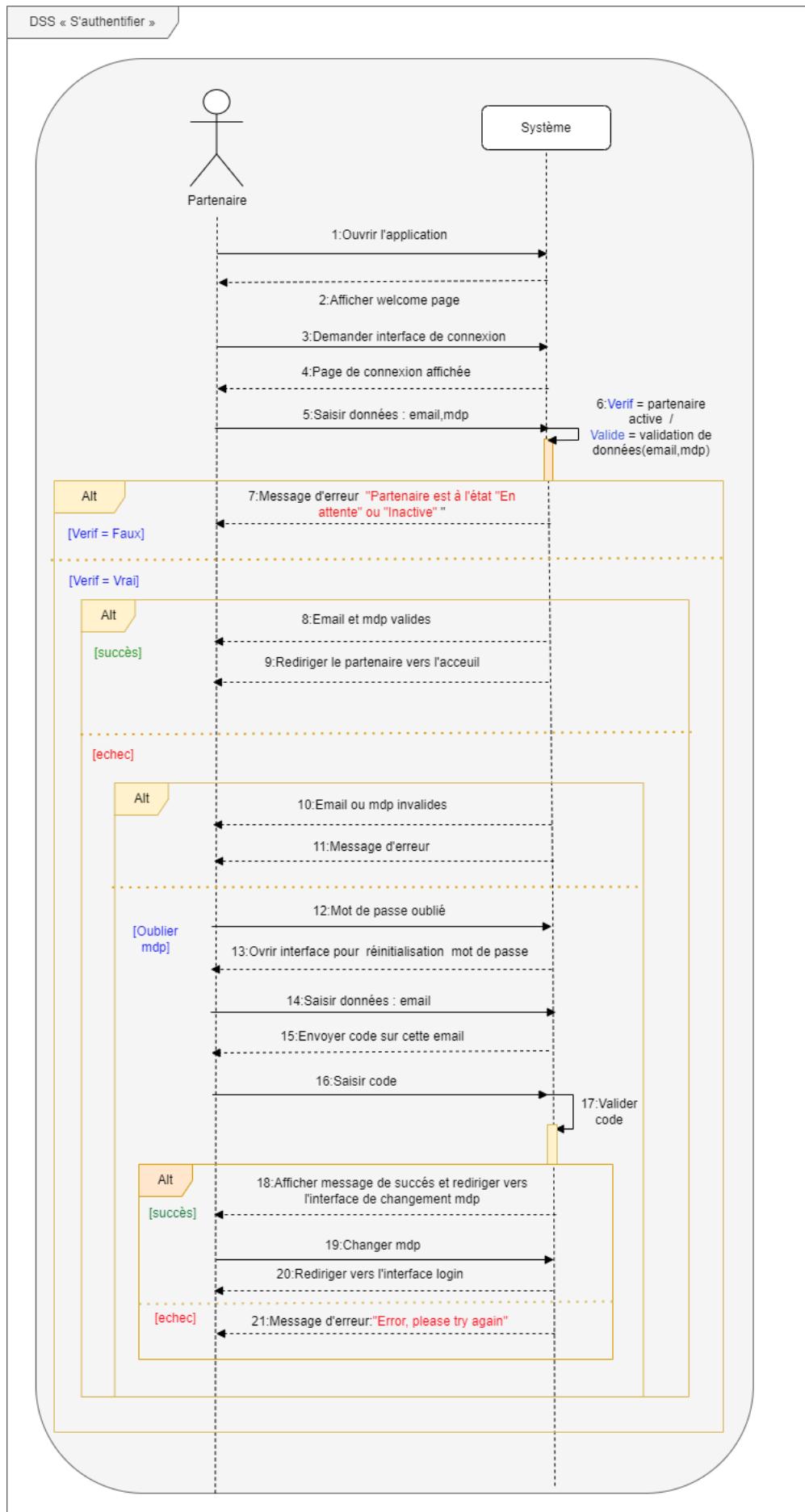


FIGURE 2.8 – Diagramme de séquence système « S'authentifier »

La description de scénarios des cas d'utilisation : « S'authentifier » est donnée par le tableau 2.6 :

Titre	S'authentifier
Acteurs	Partenaire.
Objectif	Afin de bénéficier des services proposés par notre application, une authentification est nécessaire pour tous les utilisateurs de notre application.
Précondition	Partenaire possède déjà un compte.
Post-condition	Partenaire authentifié et activé.
Scénario nominal	<p>[S1] :Authentification simple :</p> <ul style="list-style-type: none"> (a) Le partenaire demande l'interface d'authentification. (b) Le système affiche le formulaire. (c) L'utilisateur saisit son email et mot de passe, puis valide. (d) Le système vérifie les champs de partenaire et lui attribue un jeton d'authentification.[A1] <p>[S2] :Oublier mot de passe :</p> <ul style="list-style-type: none"> (a) Le partenaire clique sur "oublier mot de passe". (b) Le système affiche le champ d'email. (c) Le partenaire saisit son email. (d) Le système vérifie si c'est une adresse e-mail appartenant à l'un de ces partenaires, puis envoie un code à cette adresse e-mail.[A1] (e) Le partenaire saisit le code. (f) Le système vérifie le code et affiche le champ de changement de mot de passe.[A2] (g) Le partenaire modifie son mot de passe et le système le redirige vers l'authentification.
Scénario alternatif	<ul style="list-style-type: none"> — [S1.A1.a] : L'email ou le mot de passe saisis sont incorrects. — [S1.A1.b] : Le partenaire n'est pas encore validé par l'administrateur ou il est inactive. — [S2.A1] :L'e-mail n'appartient à aucun des partenaires. — [S2.A2] :Le code saisi est incorrect,ré-envoyer l'email.
Exigences supplémentaires	<ul style="list-style-type: none"> — L'interface doit être ergonomique. — Les messages doivent être compréhensibles et clairs.

TABLE 2.6 – Description textuelle du cas « S'authentifier »

2.4.2.4. Diagramme de séquence système « Gérer profil » :

La figure 2.9 présente le diagramme de séquence système gérer profil.

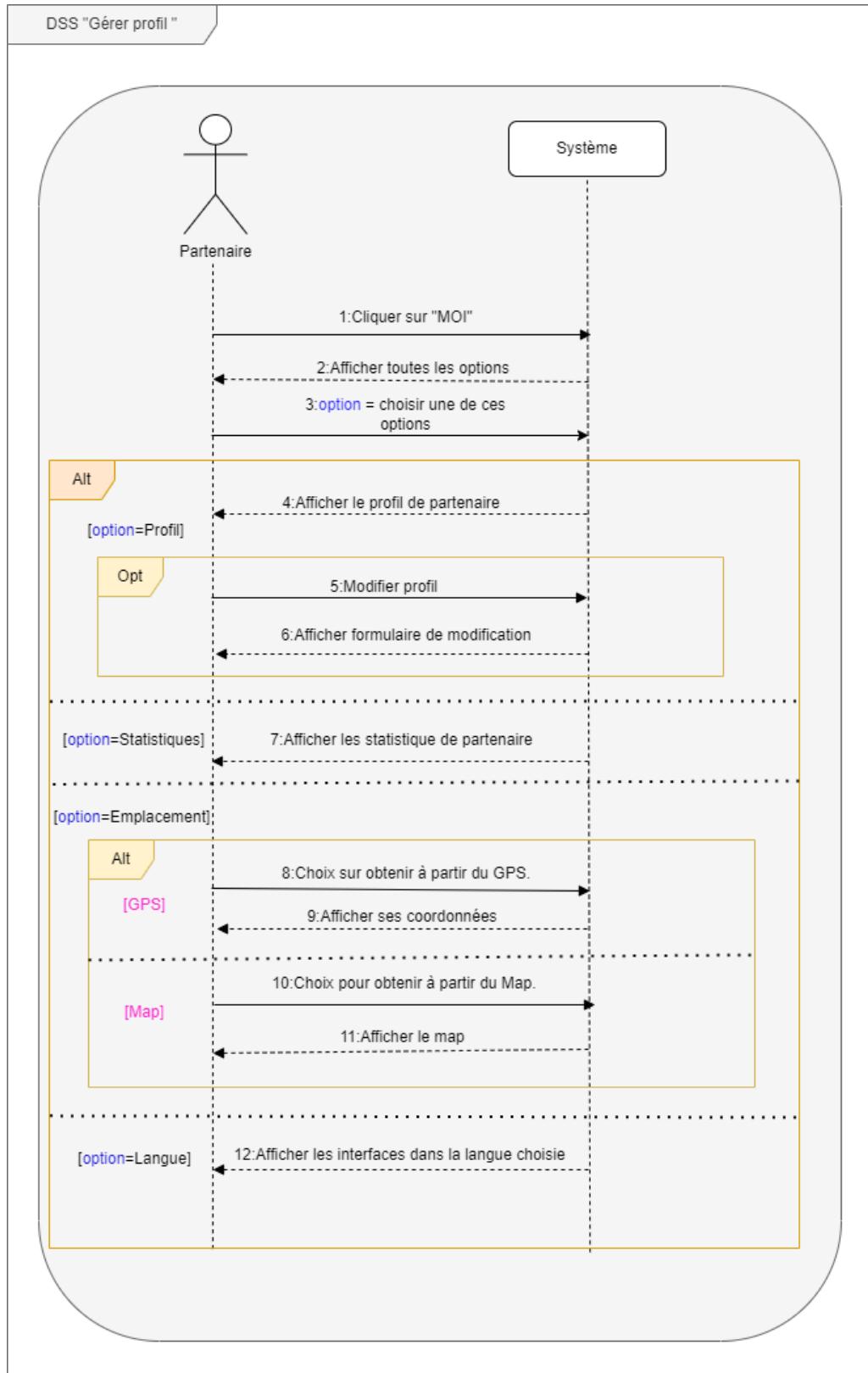


FIGURE 2.9 – Diagramme de séquence système « Gérer profil »

La description de scénarios des cas d'utilisation : « Gérer profil » est donnée par le tableau 2.7 :

Titre	Gérer profil
Acteurs	Partenaire.
Objectif	La gestion des informations personnelles avec la possibilité de les modifier. Les statistiques de revenus peuvent être consultées et la langue de l'application peut être modifiée.
Précondition	Partenaire authentifié.
Post-condition	Personnalisation du profil partenaire.
Scénario nominal	<p>[S1] :Gérer son profil :</p> <ul style="list-style-type: none"> (a) Le partenaire clique sur "Profil". (b) Le système affiche tous les informations de partenaire. (c) Le partenaire peut cliquer sur "Modifier votre profil". (d) Le système affiche formulaire rempli par les informations de partenaire. (e) Le partenaire peut modifier ses informations, son image ou son mot de passe, puis les enregistrer. (f) Le système vérifie les champs, affiche un message de succès et redirige vers le profil.[A1] <p>[S2] :Changer son adresse :</p> <ul style="list-style-type: none"> (a) Le partenaire clique sur "Emplacement". (b) Le système affiche une liste des options. (c) Le partenaire peut choisir soit d'obtenir sa position à partir du signal GPS, soit de l'obtenir à partir d'une carte géographique. (d) En cas d'utilisation du GPS, le système affiche une fenêtre contextuelle contenant les coordonnées, tandis que dans le cas de l'utilisation d'une carte géographique, le système affiche la carte correspondante. (e) Dans les deux cas, le partenaire modifie son adresse, le système l'affiche dans les paramètres de compte.
Scénario alternatif	— [S1.A1] : L'un des champs a été rempli de manière incorrecte, ce qui a entraîné l'affichage d'un message d'erreur par le système.
Exigences supplémentaires	<ul style="list-style-type: none"> — L'interface doit être ergonomique. — Les messages doivent être compréhensibles et clairs.

TABLE 2.7 – Description textuelle du cas « Gérer profil »

2.4.2.5. Diagramme de séquence système « Gérer boxes » :

La figure 2.10 présente le diagramme de séquence système gérer boxes.

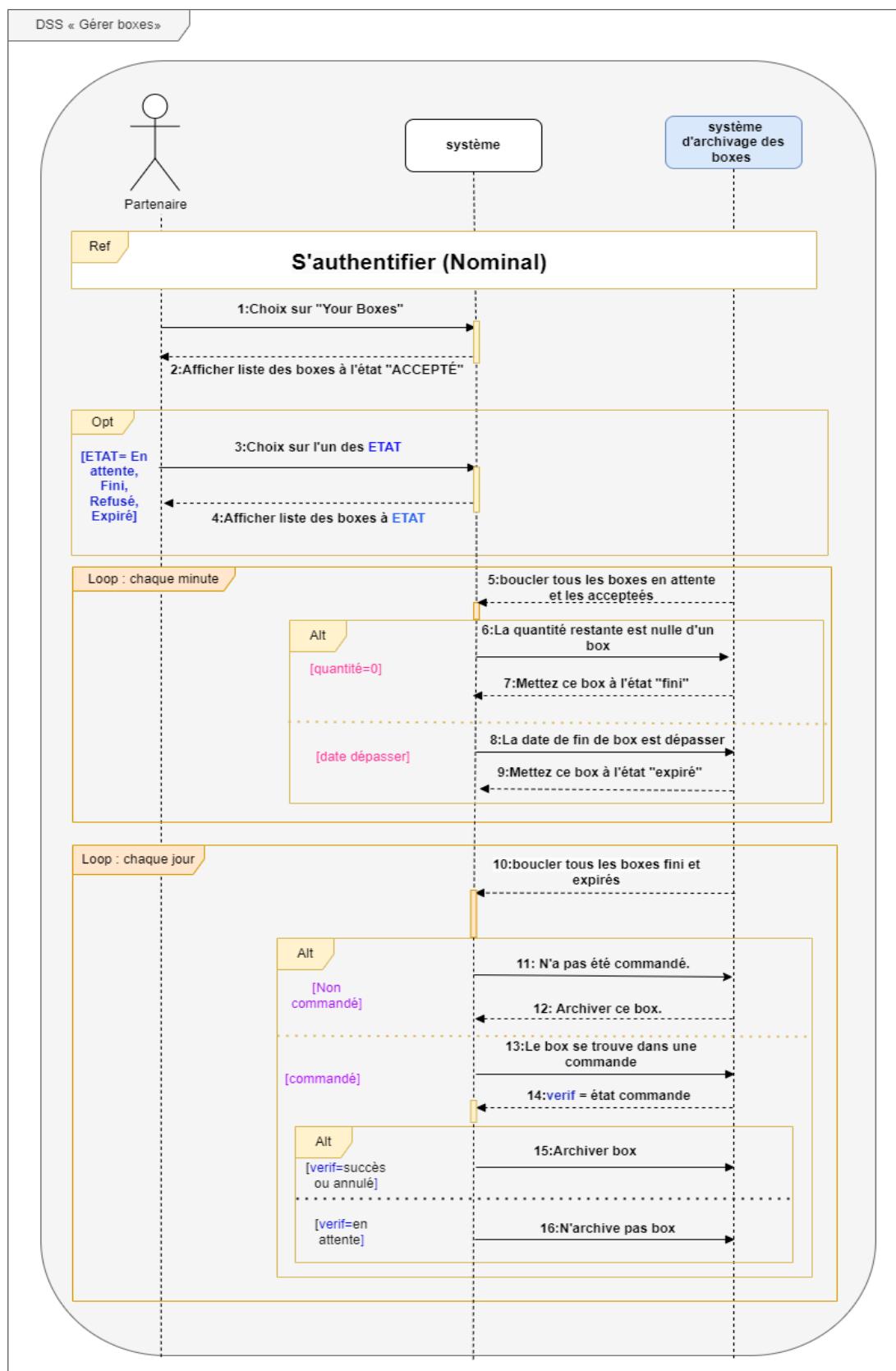


FIGURE 2.10 – Diagramme de séquence système « Gérer boxes »

La description de scénarios des cas d'utilisation : « Gérer boxes » est donnée par le tableau 2.8 :

Titre	Gérer boxes
Acteurs	Partenaire
Objectif	Gérer l'ajout et la consultation en fonction de leur état, avec la possibilité de Stocker les boxes dont la fonctionnalité se termine dans l'application.
Précondition	Partenaire authentifié et box non archivé.
Post-condition	box consulté ou archivé avec succès.
Scénario nominal	<p>[s1] : Changer les états des boxes : Le système d'archivage parcourt tous les boxes dans les états "en attente" et "acceptée". Si la quantité est nul, il les transfère à l'état "fini". De plus, s'il dépasse la date de fin, il les met à l'état "expiré". Ce processus se répète chaque minute.</p> <p>[s2] : Archiver les boxes : Le système d'archivage parcourt tous les boxes à l'état finalisé et expiré. Si le box ne se trouve dans aucune commande, le système archive ce box. Sinon, s'il se trouve dans l'une des commandes, il vérifie l'état de la commande. Si son état est "succès" ou "annulé", il archive le box. Sinon, si le box se trouve dans une commande à l'état en attente, il le laisse tel quel.</p>
Exigences supplémentaires	— Ce système d'archivage fonctionne automatiquement.

TABLE 2.8 – Description textuelle du cas «Archiver boxes»

2.4.2.6. Diagramme de séquence système « Consulter commandes » :

La figure 2.11 présente le diagramme de séquence système consulter commandes.

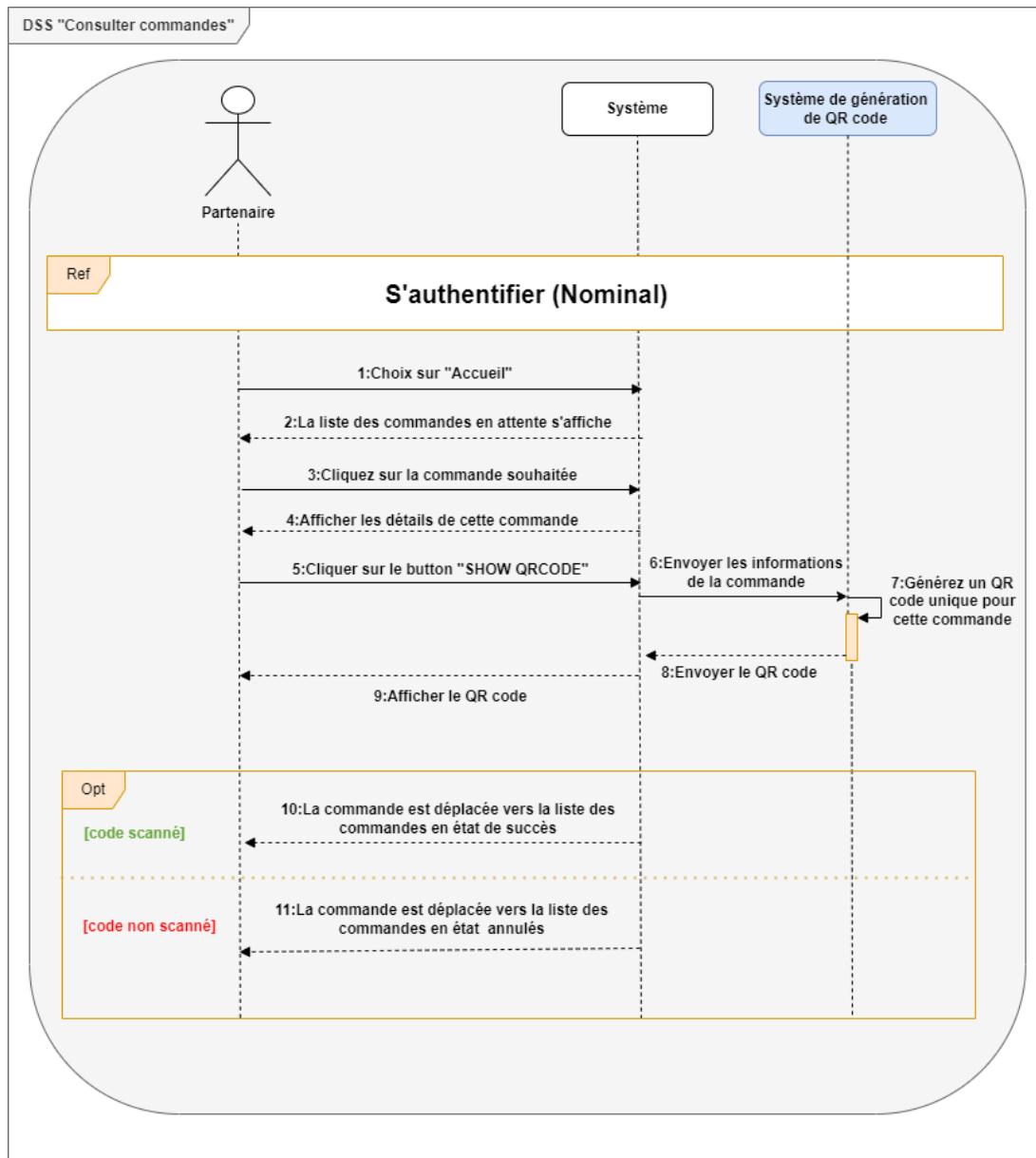


FIGURE 2.11 – Diagramme de séquence système « Consulter commandes »

La description de scénarios des cas d'utilisation : « Consulter commandes» est donnée par le tableau 2.9 :

Titre	Consulter Commandes
Acteurs	Partenaire.
Objectif	Le partenaire peut consulter les commandes et valider celles en attente. Il peut consulter les commandes réussies et annulées.
Précondition	Partenaire authentifié.
Post-condition	Commande consultée avec succès.

Scénario nominal	<ul style="list-style-type: none"> (a) Le partenaire fait son choix sur "Accueil". (b) Le système affiche la liste des commandes à l'état en attente.[A1] (c) Le partenaire peut sélectionner l'une des commandes dans la liste. (d) Le système affiche les détails de la commande sélectionnée. (e) Le partenaire clique sur le bouton "SHOW QR CODE". (f) Le système génère un QR code pour cette commande et l'affiche. (g) Si le QR code est scanné par le client, le système affiche la commande comme "réussie". Sinon, si la date de fin de la commande est dépassée et que le code n'est pas scanné, le système affiche la commande comme "annulée".
Scénario alternatif	<ul style="list-style-type: none"> — [A1] : La liste des commandes est vide.
Exigences supplémentaires	<ul style="list-style-type: none"> — L'interface doit être ergonomique.

TABLE 2.9 – Description textuelle du cas « Consulter commandes »

2.4.3 Sprint 3 : Partie mobile côté client

2.4.3.1 Backlog du sprint 3

Le tableau 2.10 décrit le Backlog du produit de notre partie mobile côté client :

Numéro	User Story	Acteur	Tâches	Durée	Priorité
1	Authentification et inscription	Client	<ul style="list-style-type: none"> — Incrire client. — Connexion client. 	4	1
2	Gérer profil	Client	<ul style="list-style-type: none"> — Consulter et modifier son profil. — Consulter ses statistiques. 	4	2

3	Procédure de réservations	Client	<ul style="list-style-type: none"> — Consulter boxes. — Consulter les détails de boxes et de ses partenaires. — Effectuer des réservations. — Valider les commandes. 	12	1
4	Ajout de fonctionnalités avancées	Client	<ul style="list-style-type: none"> — Ajouter des boxes et des partenaires aux favoris. — Supprimer les boxes et les partenaires de la liste des favoris. — Filtrer des boxes. — chercher des boxes. — Ajouter des fonctionnalités basées sur la géolocalisation. 	12	1

**TABLE 2.10 – Backlog du produit de partie mobile côté client
de la partie**

2.4.3.2. Diagramme de navigation système (DNS) de sprint 3 :

La figure 2.12 représente les différents pages qui composent la partie mobile côté client, les transitions entre ces écrans ainsi que les informations sur les actions disponibles à partir de chaque écran.

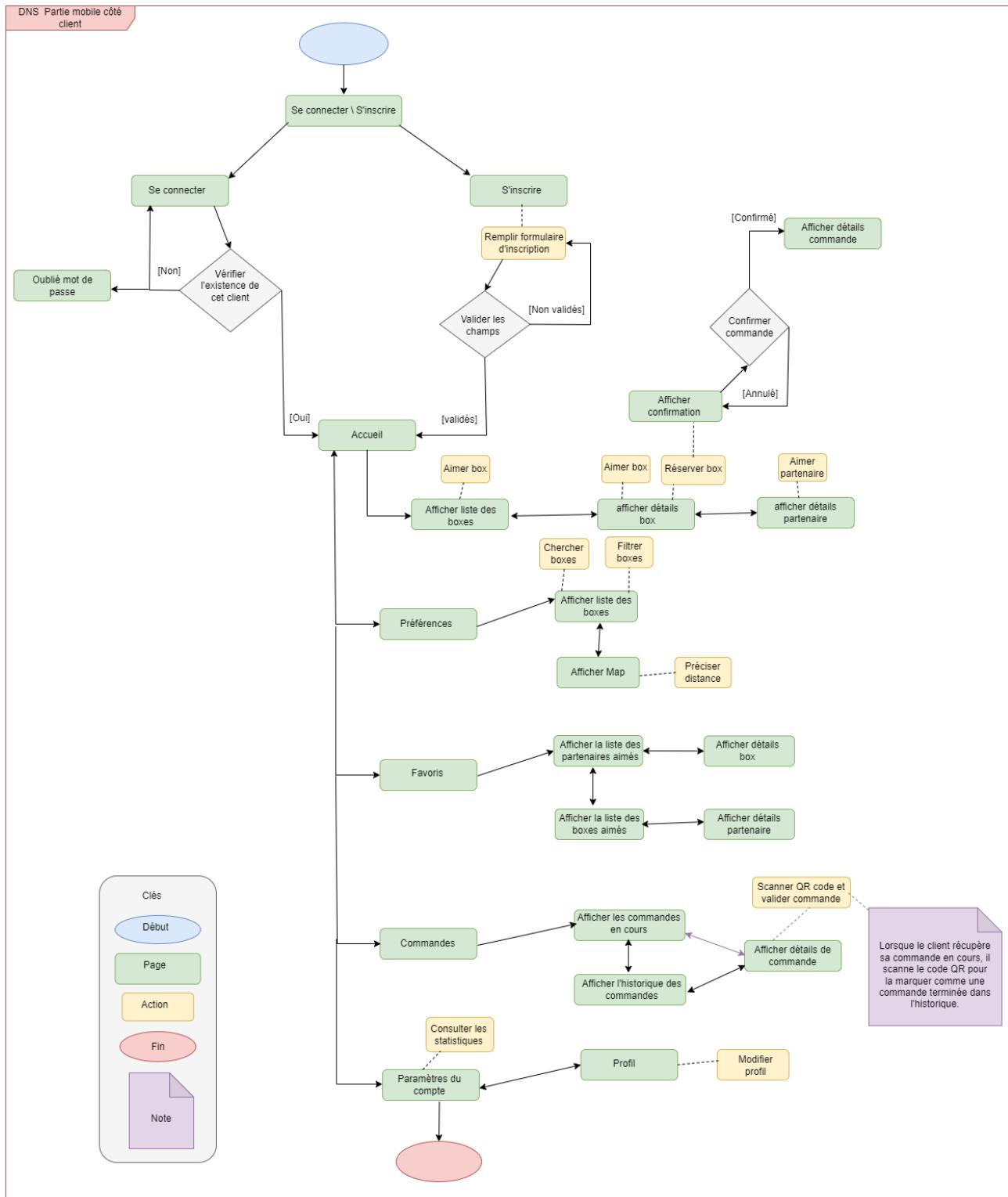


FIGURE 2.12 – Diagramme de navigation de partie mobile côté client
la partie

2.4.3.3. Raffinement du cas d'utilisation « Réserver box » :

La figure 2.13 présente le raffinement de cas d'utilisation réserver box, suivie par leur description textuelle.

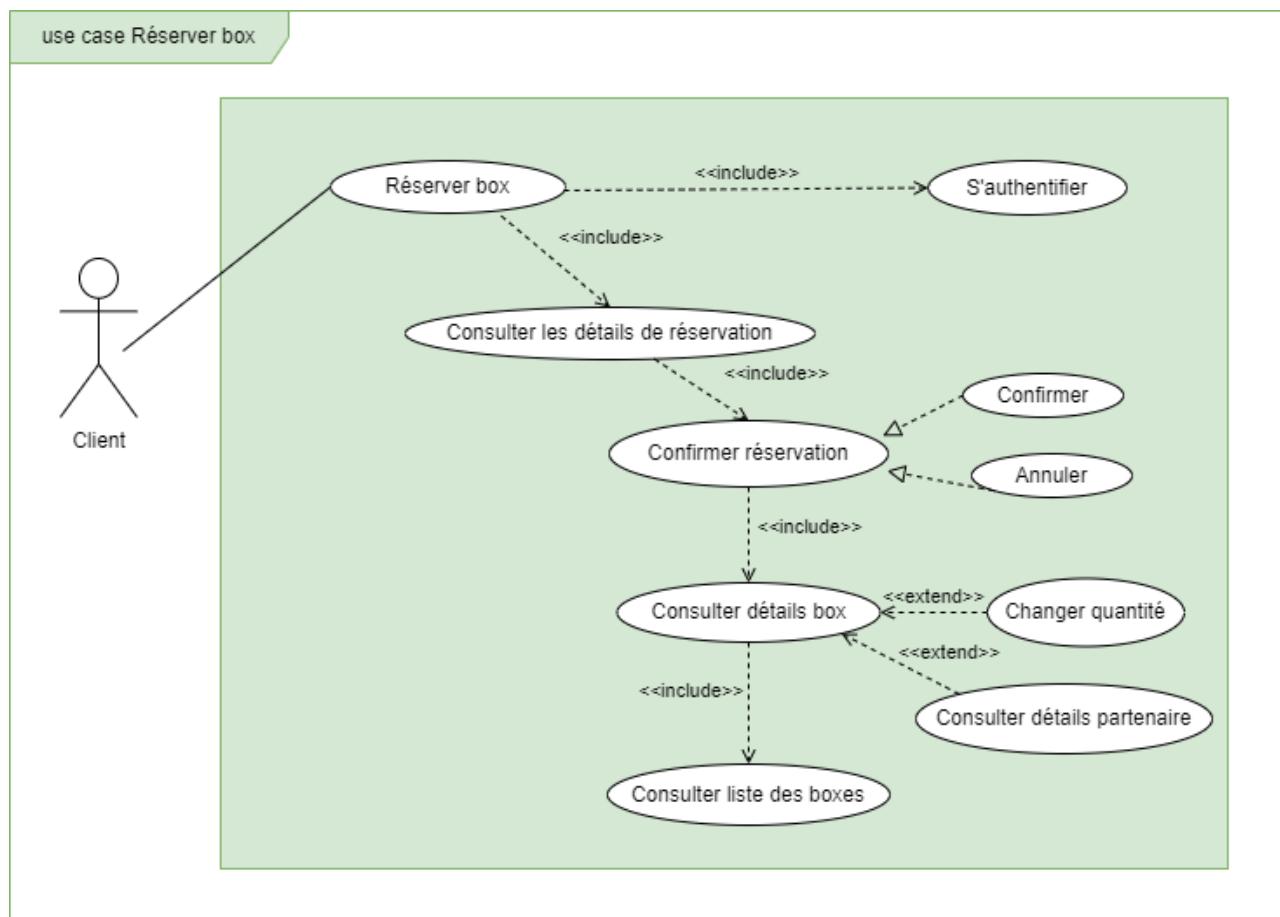


FIGURE 2.13 – Diagramme détaillé de cas d'utilisation « Réserver box »

La description de scénarios des cas d'utilisation : « Réserver box » est donnée par le tableau 2.11 :

Titre	Réserver box
Acteurs	Client
Objectif	Cette fonctionnalité vise à offrir à l'utilisateur la possibilité de réserver un box spécifique selon ses besoins.
Précondition	Client authentifié.
Post-condition	Box réservé et commande passée avec succès.
Scénario nominal	<ul style="list-style-type: none"> (a) Le client fait son choix sur "Accueil". (b) Le système affiche la liste des boxes disponibles.[A1] (c) Le client sélectionne le box qu'il souhaite. Les détails de ce box s'affichent, et il peut modifier la quantité avant de réserver ou avant de passer la commande (d) Le système affiche une fenêtre contextuelle pour confirmer la réservation ou l'annuler. (e) Le client confirme la réservation. (f) Le système affiche les details de cette réservation.

Scénario alternatif	— [A1] : La listes de boxes est vide.
Exigences supplémentaires	<ul style="list-style-type: none"> — L'interface doit être ergonomique. — Les messages doivent être compréhensibles et clairs. — Lorsque vous choisissez la quantité, elle ne peut pas dépasser la quantité disponible.

TABLE 2.11 – Description textuelle du cas « Réserver box »

2.4.3.4. Raffinement du cas d'utilisation « Valider commandes » :

La figure 2.14 présente le raffinement de cas d'utilisation valider commandes, suivie par leur description textuelle.

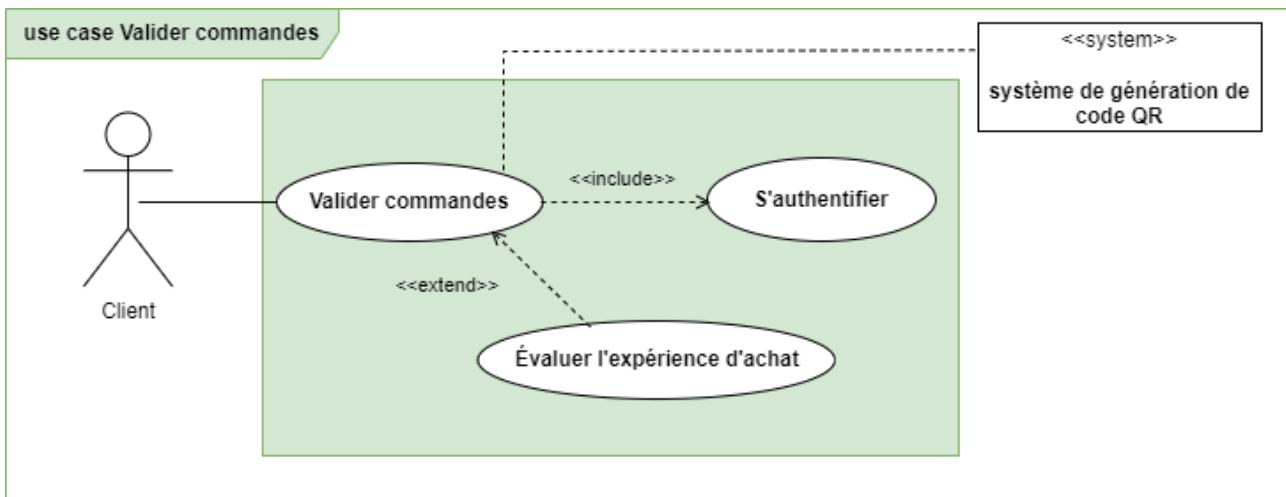


FIGURE 2.14 – Diagramme détaillé de cas d'utilisation « Valider commandes »
du

La description de scénarios des cas d'utilisation : « Valider commandes » est donnée par le tableau 2.12 :

Titre	Valider commandes
Acteurs	Client
Objectif	Cette fonctionnalité a pour objectif de permettre à l'utilisateur de récupérer sa commande avec succès.
Précondition	Client authentifié.
Post-condition	Commande récupérée avec succès.

Scénario nominal	<ul style="list-style-type: none"> (a) Le client fait son choix sur "Orders". (b) Le système affiche la liste des commandes en cours.[A1] (c) Le client sélectionne la commande en cours de récupération. (d) Le système affiche les détails de cette commande. (e) Le client clique sur "SCANNER QR CODE". (f) Le système active la caméra pour scanner le code QR. (g) Le code QR est scanné avec succès, et la commande est désormais enregistrée dans l'historique. (h) Le système affiche une interface d'évaluation. (i) Le client choisit d'évaluer ou de passer cette interface.
Scénario alternatif	<ul style="list-style-type: none"> — [A1] : La liste des commandes en cours est vide.
Exigences supplémentaires	<ul style="list-style-type: none"> — L'interface doit être ergonomique. — Les messages doivent être compréhensibles et clairs.

TABLE 2.12 – Description textuelle du cas « Valider commandes »

2.4.3.5. Raffinement du cas d'utilisation « Sélectionner boxes» :

La figure 2.15 présente le raffinement de cas d'utilisation sélectionner boxes, suivie par leur description textuelle.

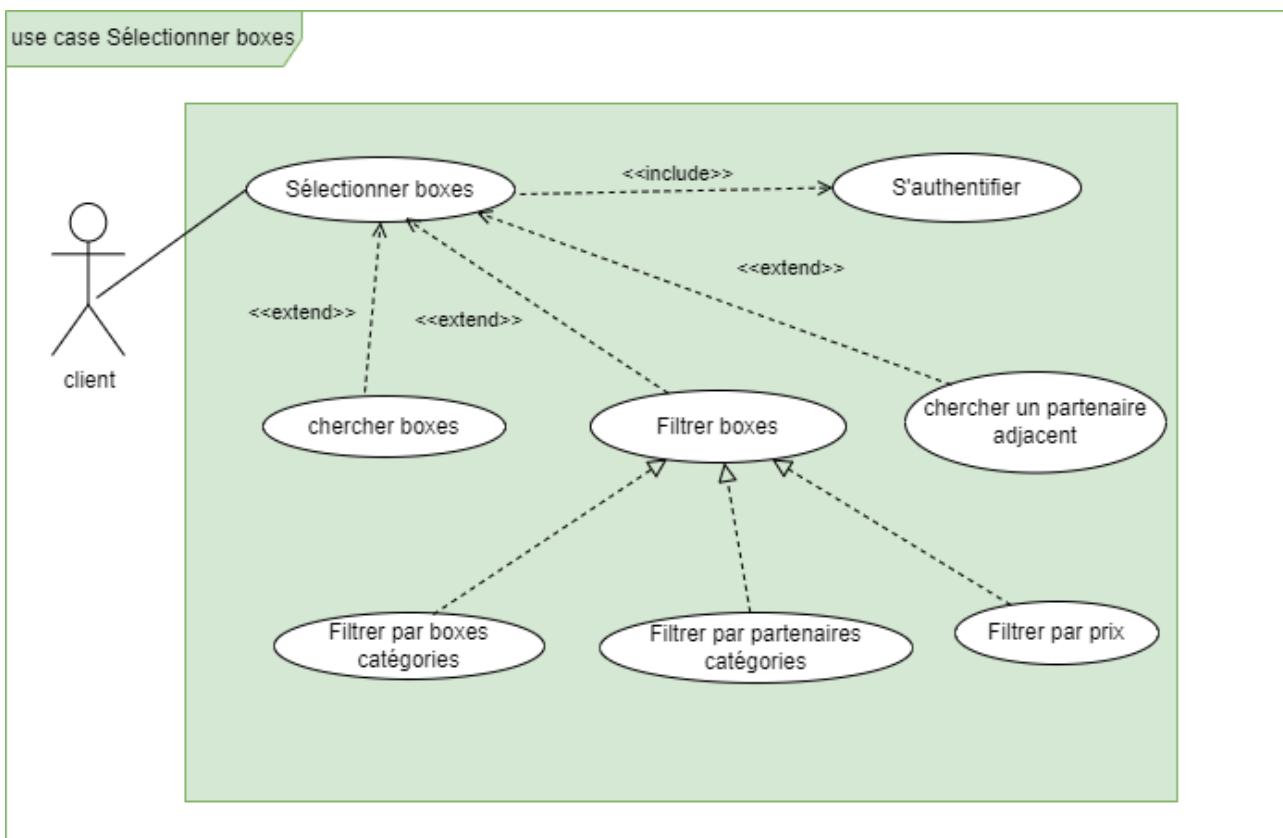


FIGURE 2.15 – Diagramme détaillé de cas d'utilisation « Sélectionner boxes» du

La description de scénarios des cas d'utilisation : « Sélectionner boxes» est donnée par le tableau 2.13 :

Titre	Sélectionner boxes
Acteurs	Client
Objectif	Fournir aux clients une solution pratique pour trouver et choisir des boxes qui correspondent à leurs préférences, ainsi que pour localiser les partenaires à proximité.
Précondition	Client authentifié.
Post-condition	Filtrage des boxes ou bien localisation des partenaires.

Scénario nominal	<p>[S1] : Filtrer les boxes :</p> <ul style="list-style-type: none"> (a) Le client fait son choix sur "Preferences". (b) Le système affiche "List". (c) Le client sélectionne l'icône de filtre en cliquant dessus. (d) Le système affiche une fenêtre contenant tous les critères pouvant être modifiés. (e) Le client modifie ses critères selon ses besoins. (f) Le système affiche la liste des boxes disponibles correspondant à ses critères.[A1] <p>[S2] : Localisation les partenaires :</p> <ul style="list-style-type: none"> (a) Le client fait son choix sur "Cart". (b) Le système affiche l'emplacement actuel du client sur la carte géographique. (c) Le client ajuste la distance de recherche sur la carte. (d) Le système affiche les partenaires situés dans cette distance spécifique.[A1]
Scénario alternatif	<ul style="list-style-type: none"> — [S1.A1] : La liste est vide, aucun box disponible ne correspond à ses critères. — [S2.A1] : Aucun marqueur n'est affiché, il n'y a pas de partenaire dans cette zone.
Exigences supplémentaires	<ul style="list-style-type: none"> — L'interface doit être ergonomique.

TABLE 2.13 – Description textuelle du cas « Sélectionner boxes»

2.4.3.6. Raffinement du cas d'utilisation « Gérer les favoris» :

La figure 2.16 présente le raffinement de cas d'utilisation gérer les favoris, suivie par leur description textuelle.

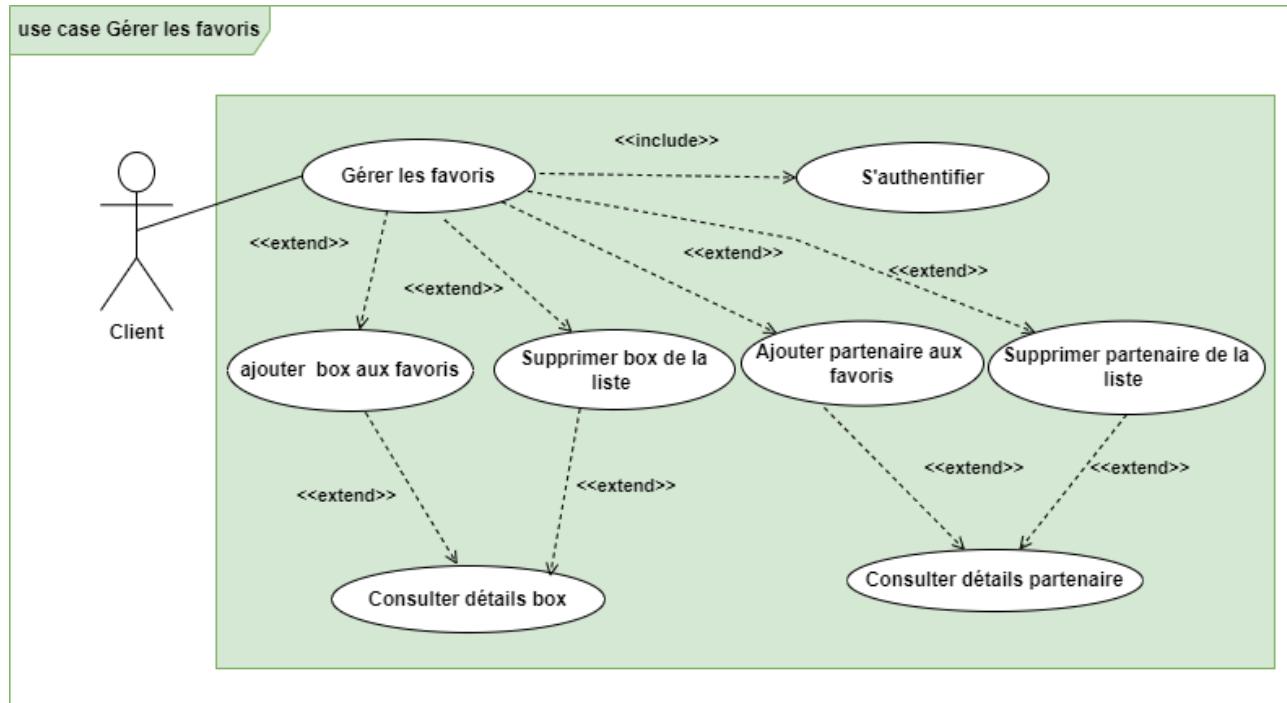


FIGURE 2.16 – Diagramme détaillé de cas d'utilisation « Gérer les favoris »

La description de scénarios des cas d'utilisation : « Gérer les favoris » est donnée par le tableau 2.14 :

Titre	Gérer les favoris
Acteurs	Client
Objectif	Offrir aux clients la fonctionnalité d'ajouter ou de supprimer des partenaires et des boxes de leur liste de favoris.
Précondition	Client authentifié.
Post-condition	Le box a été ajouté aux favoris avec succès ou bien le partenaire.

Scénario nominal	<p>[S1] : Ajouter box aux favoris :</p> <ul style="list-style-type: none"> (a) Le client fait son choix parmi les boxes affichées. (b) Le système affiche les détails de box sélectionnée. (c) Le client aime ce box en cliquant sur le cœur.[A1] (d) Le système ajoute ce box à la liste des favoris des boxes. <p>[S2] : Ajouter partenaire aux favoris :</p> <ul style="list-style-type: none"> (a) Le client fait son choix parmi les boxes affichées. (b) Le système affiche les détails de box sélectionnée. (c) Le client consulte les informations du partenaire de ce box. (d) Le système affiche les détails de ce partenaire. (e) Le client aime ce partenaire en cliquant sur le cœur.[A1] (f) Le système ajoute ce partenaire à la liste des favoris des partenaires.
Scénario alternatif	— [S1.A1] [S2.A1] : Ce box ou ce partenaire se trouve déjà dans la liste des favoris.
Exigences supplémentaires	— Le cœur sur lequel le client clique pour exprimer son appréciation est rouge lorsque le box ou le partenaire est en favoris, et il est gris dans le cas contraire.

TABLE 2.14 – Description textuelle du cas « Gérer les favoris »

Conclusion

Dans ce chapitre, nous avons commencé par identifier les principaux acteurs de notre système ainsi que leurs rôles spécifiques. Ensuite, nous avons détaillé les exigences fonctionnelles et non fonctionnelles, puis nous avons présenté les diagrammes de cas d'utilisation globaux, ainsi que leurs détails. De plus, nous avons examiné les diagrammes de navigation et les diagrammes de séquences système. Le prochain chapitre sera consacré à une vue d'ensemble conceptuelle.

Chapitre 3

Aperçu conceptuel

Introduction

Dans ce chapitre, nous allons présenter la conception de notre projet. Cela nous permettra d'assimiler correctement les besoins déjà mentionnés dans le chapitre précédent et de préparer le terrain pour la réalisation dans le chapitre suivant. Pour ces raisons, nous allons spécifier la vue architecturale, suivie de la vue comportementale de notre projet, en expliquant les patrons utilisés et en appliquant les diagrammes UML.

3.1 Architecture générale

Lors de la conception d'un système logiciel, il revêt une importance capitale de sélectionner avec soin le modèle d'architecture approprié afin d'assurer un fonctionnement harmonieux entre les diverses couches. Cette démarche garantit le bon déroulement de l'ensemble du système.

3.1.1 Les patrons d'architecture

Le patron d'architecture fournit un cadre solide pour structurer et organiser les différents composants d'un système, ce qui facilite la compréhension, la maintenance et l'évolutivité du code.

Dans notre projet, nous avons utilisé :

1. Le patron de déploiement : 3-tiers

séparer clairement

Ce patron est un modèle logique qui vise à clairement séparer trois couches logicielles au sein d'une même application. Voici une brève description des trois niveaux de cette architecture :

- **Couche présentation** : Il s'agit de la partie visible de l'application qui permet à l'utilisateur d'interagir avec le système. Il gère l'interface utilisateur, l'interaction utilisateur et l'affichage des données. **elle (la couche)**
- **Couche applicative ou couche métier** : C'est la partie qui contient la logique métier de l'application. Il s'agit du cœur fonctionnel de l'application, où les différentes règles de gestion et qui décrit les opérations que l'application effectue sur les données

en fonction des requêtes des utilisateurs réalisées à travers la couche présentation.

- **Couche accès aux données** : C'est la couche qui est responsable de la gestion des données de l'application. Ces données sont récupérées et retournées à la couche applicative pour être traitées et renvoyées à l'utilisateur.

La figure 3.1, illustre les interactions entre les différents composants de l'architecture 3-tiers.

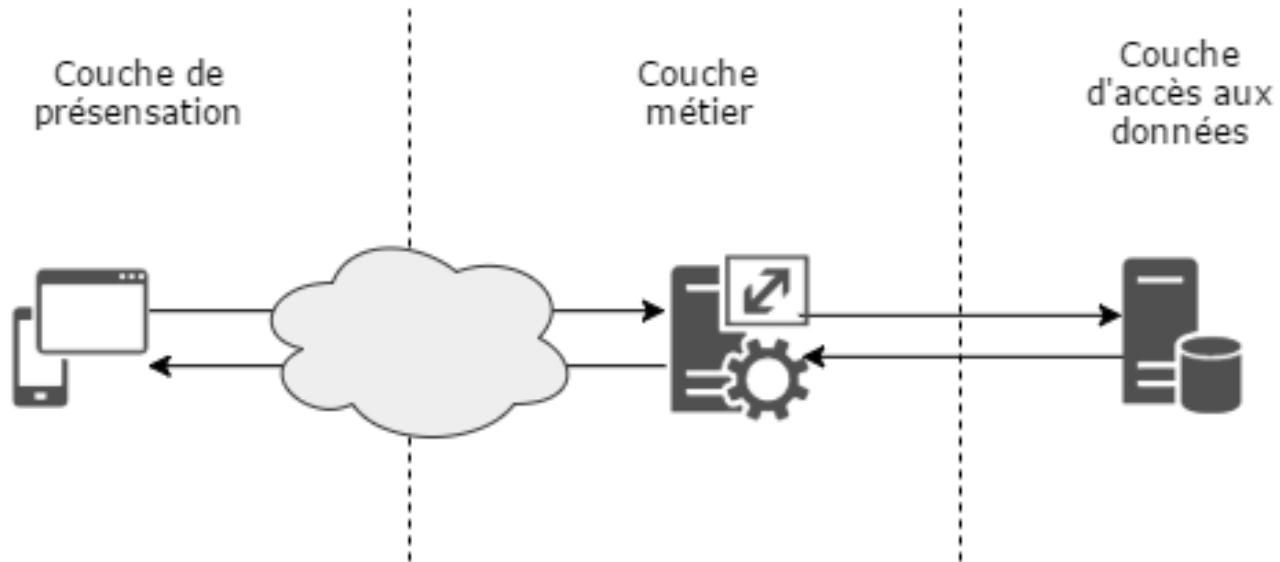


FIGURE 3.1 – Architecture 3-tiers

2. Le patron interactif : MVC (Modèle-Vue-Contrôleur)

Le patron MVC facilite la structuration efficace du code source. Elle vous guide dans la création des fichiers nécessaires tout en définissant clairement leur rôle respectif. L'objectif fondamental du modèle MVC est de diviser la logique du code en trois parties distinctes, chacune étant représentée par des fichiers séparés :

- **Modèle** : cette partie gère les données de l'application. Il contient les données brutes de l'application. Il représente une abstraction des données et fournit des méthodes pour accéder, manipuler et notifier les changements de données.
- **Vue** : cette partie se concentre sur l'affichage **des** données provenant du modèle. La vue est responsable de la présentation des informations et de leur mise en forme pour une interaction avec l'utilisateur.
- **Contrôleur** : cette partie gère la logique du code qui prend des décisions. C'est en quelque sorte l'intermédiaire entre le modèle et la vue, gérant les interactions utilisateur, coordonnant les opérations sur les données et mettant à jour l'interface utilisateur en conséquence.

La figure 3.2 présente le principe du patron MVC.

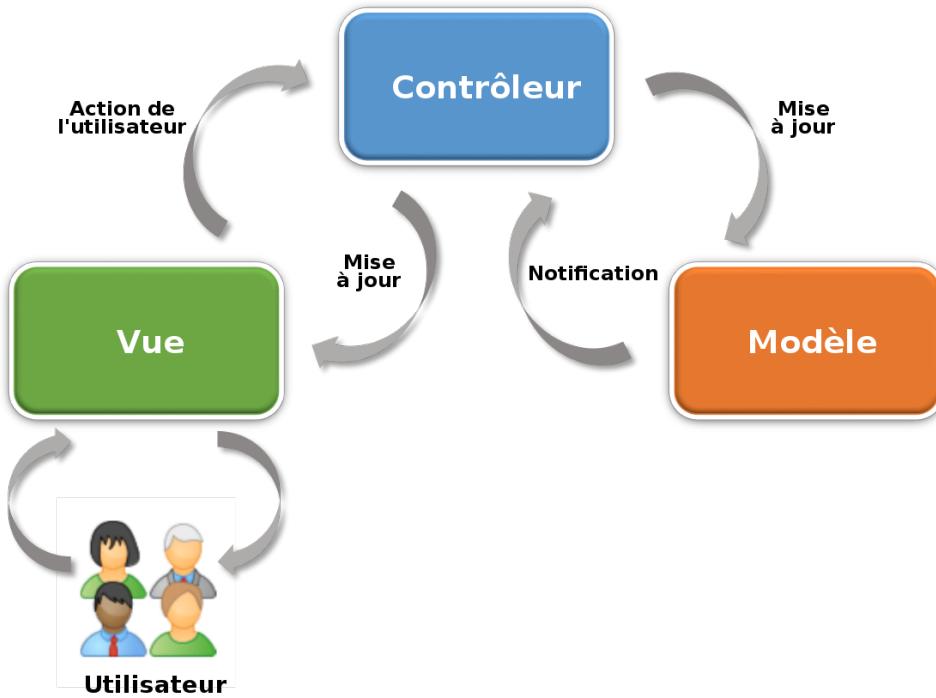


FIGURE 3.2 – Principe du patron MVC

3.1.2 Les patrons de conception

Les patrons de conception fournissent des solutions structurées pour résoudre des problèmes de conception de manière efficace. Ils permettent aussi d'améliorer la qualité du code et d'accélérer le processus de développement.

Parmi les patrons de conception dont on a fait usage durant le développement de notre application on se doit de mentionner :

1. Middleware

Un middleware permet de court-circuiter une requête de l'utilisateur. Nous pouvons, avant l'exécution de contrôleur (l'affichage de la vue, la création d'un nouveau message, la modification du mot de passe...), effectuer des tests et rediriger si nécessaire la personne.[\[11\]](#)

L'objectif du middleware est de permettre les interactions en fournissant des services unifiés aux applications tels que : L'échange des données, L'authentification,etc...

On peut citer comme exemple le Middleware que nous avons utilisé dans notre application nommé "**EnsureFrontendRequestsAreStateful**" fourni par **SANCTUM**. Ce middleware SANCTUM protège les routes API en vérifiant si les requêtes proviennent d'un client authentifié et en ajoutant le jeton d'accès aux en-têtes de la requête, permettant ainsi la gestion des jetons d'accès pour l'authentification API dans Laravel.

2. Le patron comportemental : Observer

Le patron de conception Observer fournit un mécanisme flexible pour la communication et la mise à jour des objets, en permettant aux observateurs d'être informés des changements d'état des sujets qu'ils observent.

Le but d'utiliser ce type de patron dans notre code est de gérer des actions spécifiques lors de certains événements. La figure 3.3 illustre la structure de ce patron.

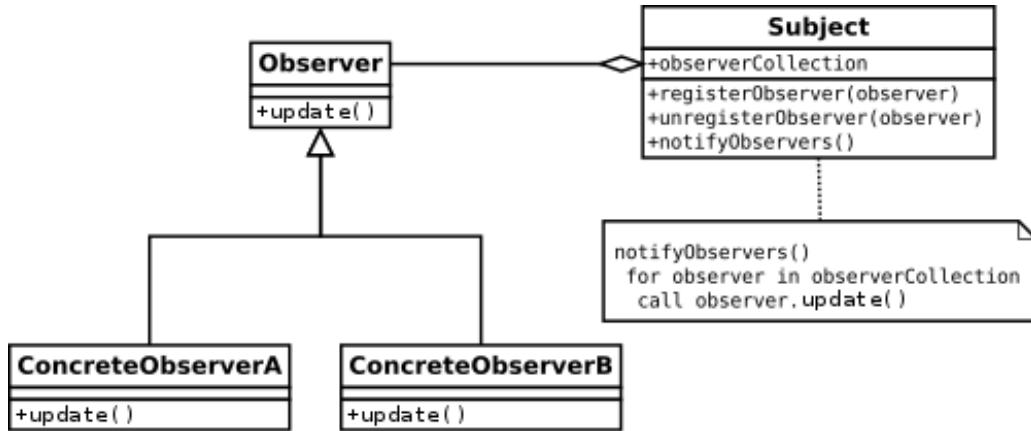


FIGURE 3.3 – Le patron de conception observer

3.2 Conception détaillée

3.2.1 Diagramme de classes des entités

Un **diagramme de classes des entités** est un type de diagramme utilisé en génie logiciel pour représenter les entités et les relations entre celles-ci dans un système informatique. Ce diagramme est considéré comme crucial dans la modélisation orientée-objet.

Il s'agit d'une représentation statique qui ne prend pas en compte les aspects temporels et dynamiques.

La figure 3.1 illustre le diagramme de classes des entités de notre projet.

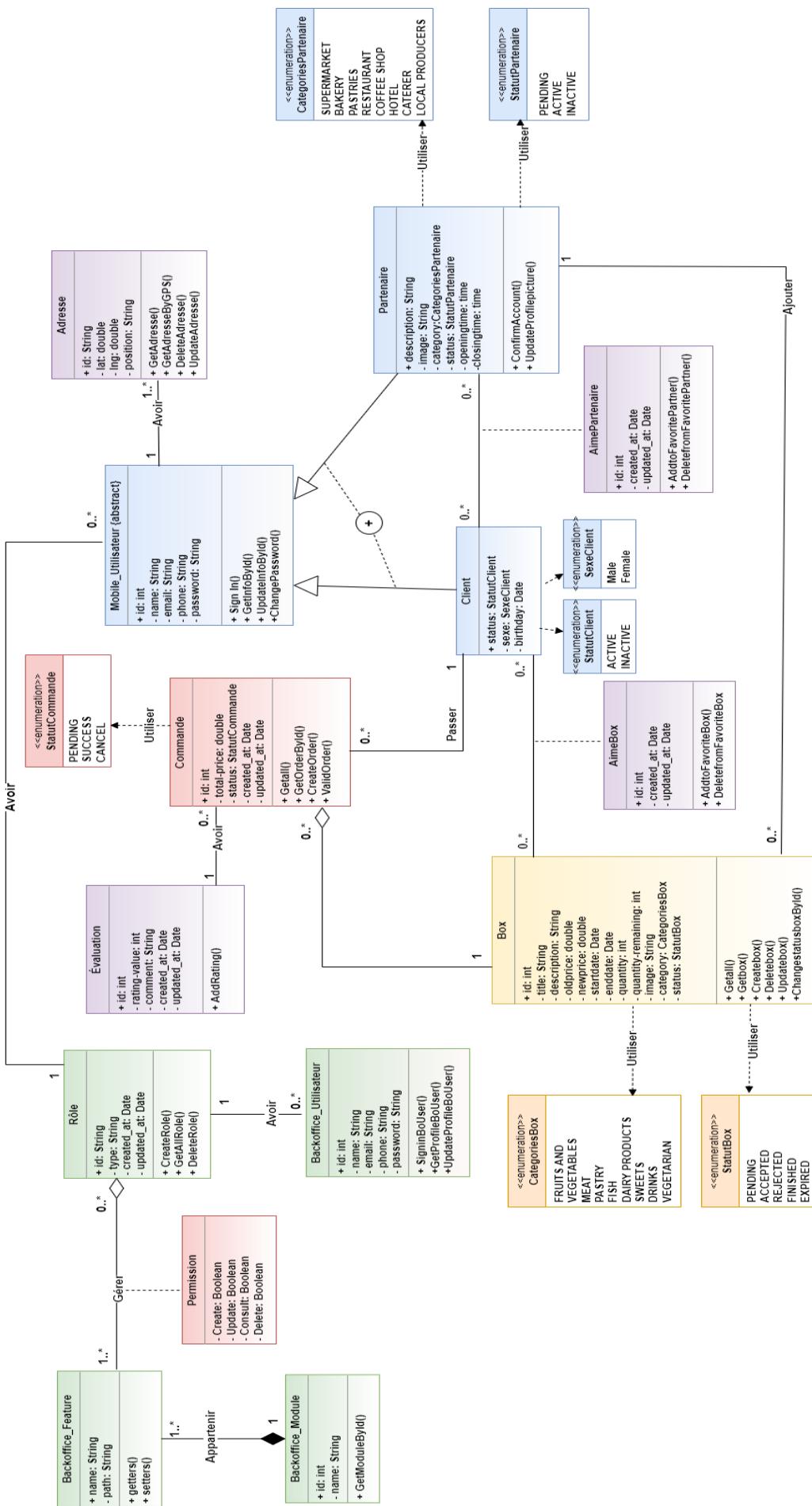


FIGURE 3.4 – Diagramme de classes des entités

Le tableau 3.1 donne une description brève de chacune des classes qui composent notre diagramme.

Classe	Description
Backoffice-Feature	Représente les fonctionnalités que l'application est capable de réaliser pour répondre aux besoins de l'utilisateur.
Backoffice-Module	Représente les modules du la partie web telque "Partners", "Boxes", etc...
Rôle	Symbolise les différents rôles que peut endosser un utilisateur afin d'accéder aux différentes fonctionnalités proposées par notre application.
Backoffice-Utilisateur	Représente l'ensemble des administrateurs qui se connectent sur le backoffice.
Mobile-Utilisateur	Représente l'ensemble des utilisateurs qui se connectent sur la partie mobile.
Client	Représente toutes les informations concernant le client.
Partenaire	Représente toutes les informations concernant le partenaire.
Commande	Détient les données relatives à une commande.
Box	Définit les informations qui concernent un box.
Évaluation	Définit les informations qui concernent l'expérience d'achat.
Adresse	Modélise les informations relatives à une adresse d'un utilisateur de la partie mobile.

TABLE 3.1 – Description des classes

3.2.2 Conception détaillée de sprint 1

3.2.2.1. Diagrammes de séquence

Un **diagramme de séquence** est un type de diagramme d'interaction, car il décrit comment et dans quel ordre plusieurs objets fonctionnent ensemble.[\[4\]](#)

3.2.2.2. DS du cas d'utilisation « Gérer les partenaires»

La figure 3.5 illustre le diagramme de séquence du cas d'utilisation « Gérer les partenaires» : Un administrateur peut modifier les informations d'un partenaire.

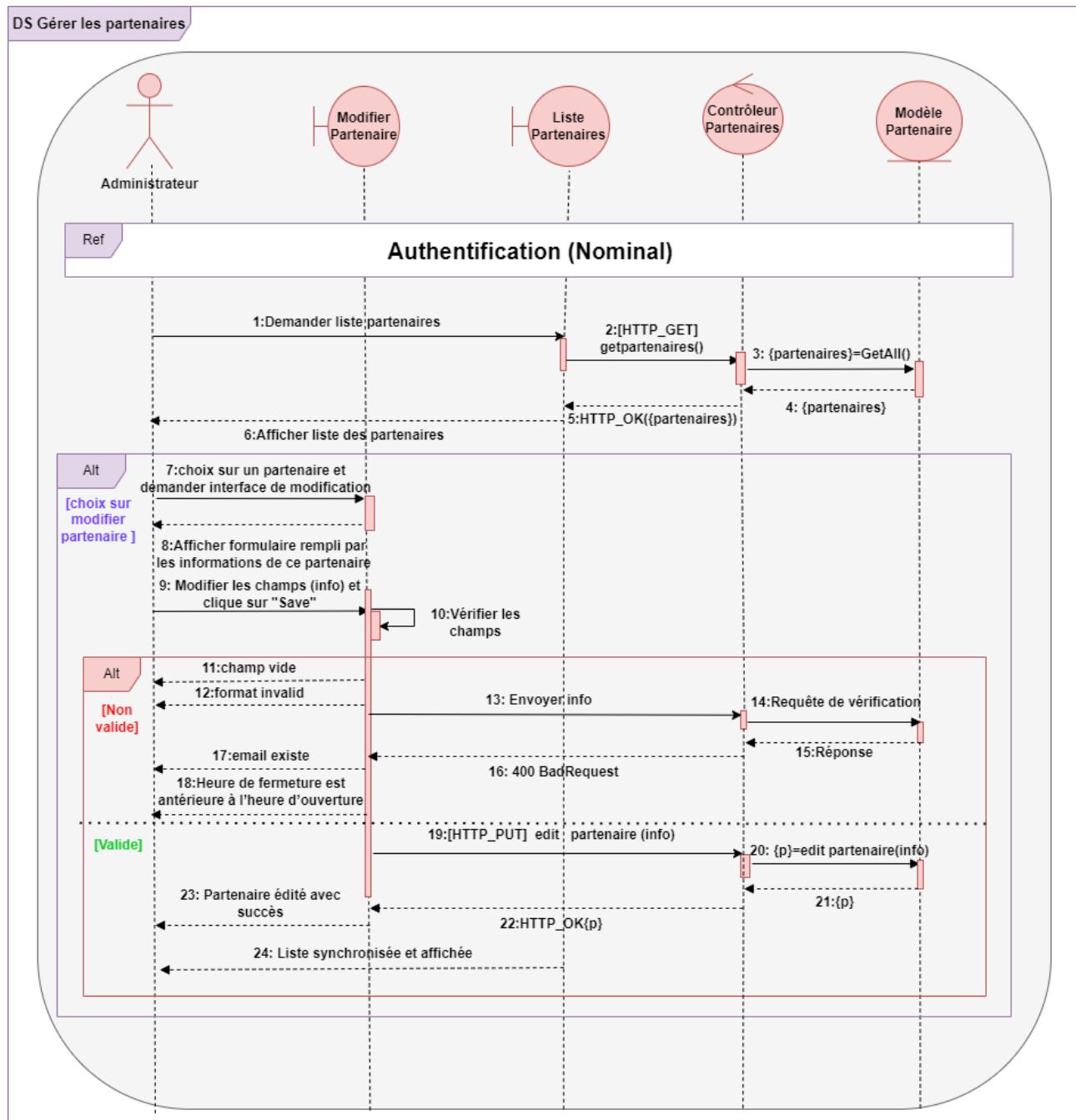


FIGURE 3.5 – DS du cas d'utilisation « Gérer les partenaires »

3.2.2.3. DS du cas d'utilisation « Gérer les clients »

La figure 3.5 illustre le diagramme de séquence du cas d'utilisation « Gérer les clients» : Un administrateur peut ajouter un client.

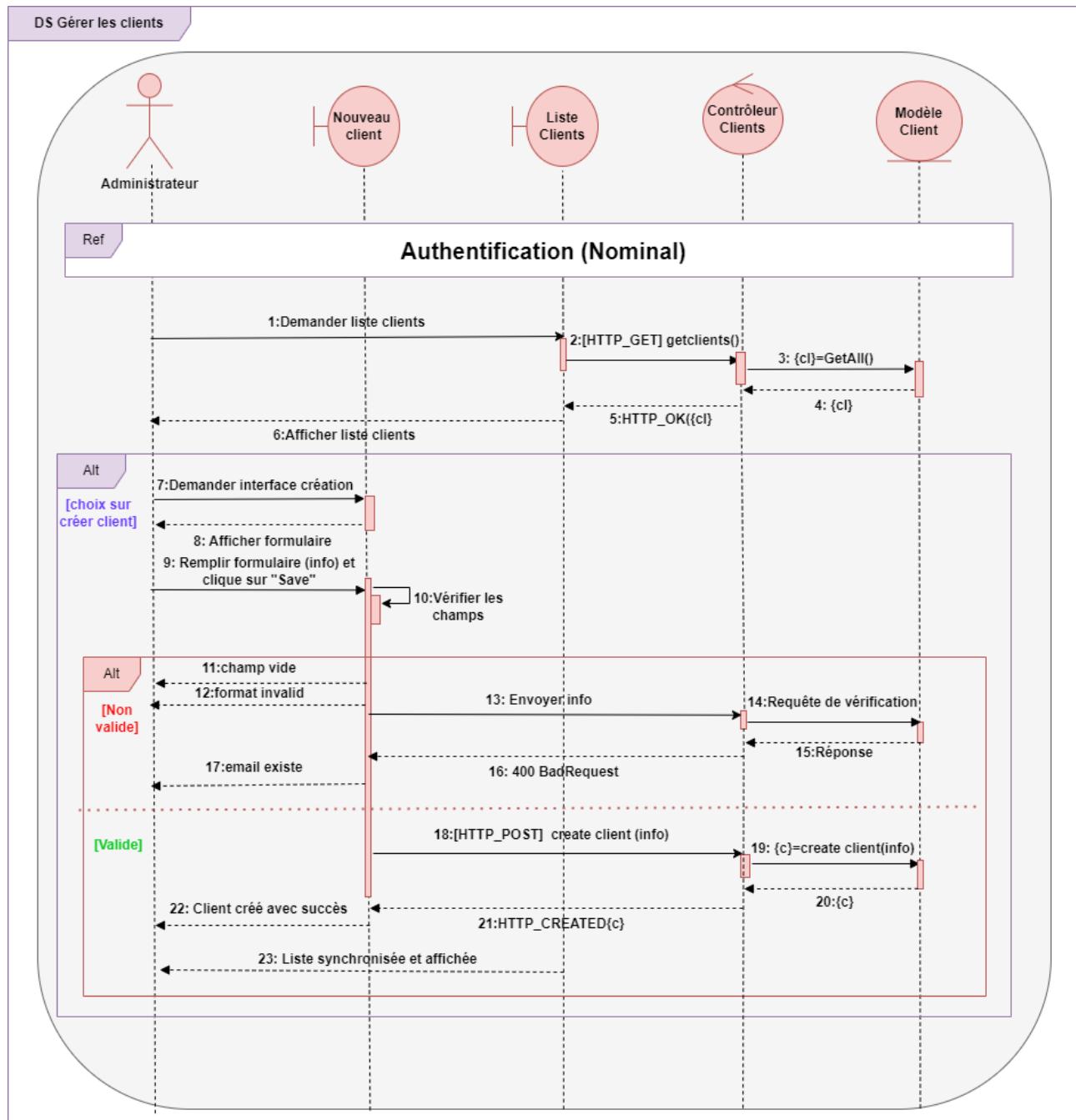


FIGURE 3.6 – DS du cas d'utilisation « Gérer les clients »

3.2.2.4. DS du cas d'utilisation « Gérer les boxes »

La figure 3.7 illustre le diagramme de séquence du cas d'utilisation « Gérer les boxes» : Un administrateur peut supprimer un box ou bien filtrer les boxes.

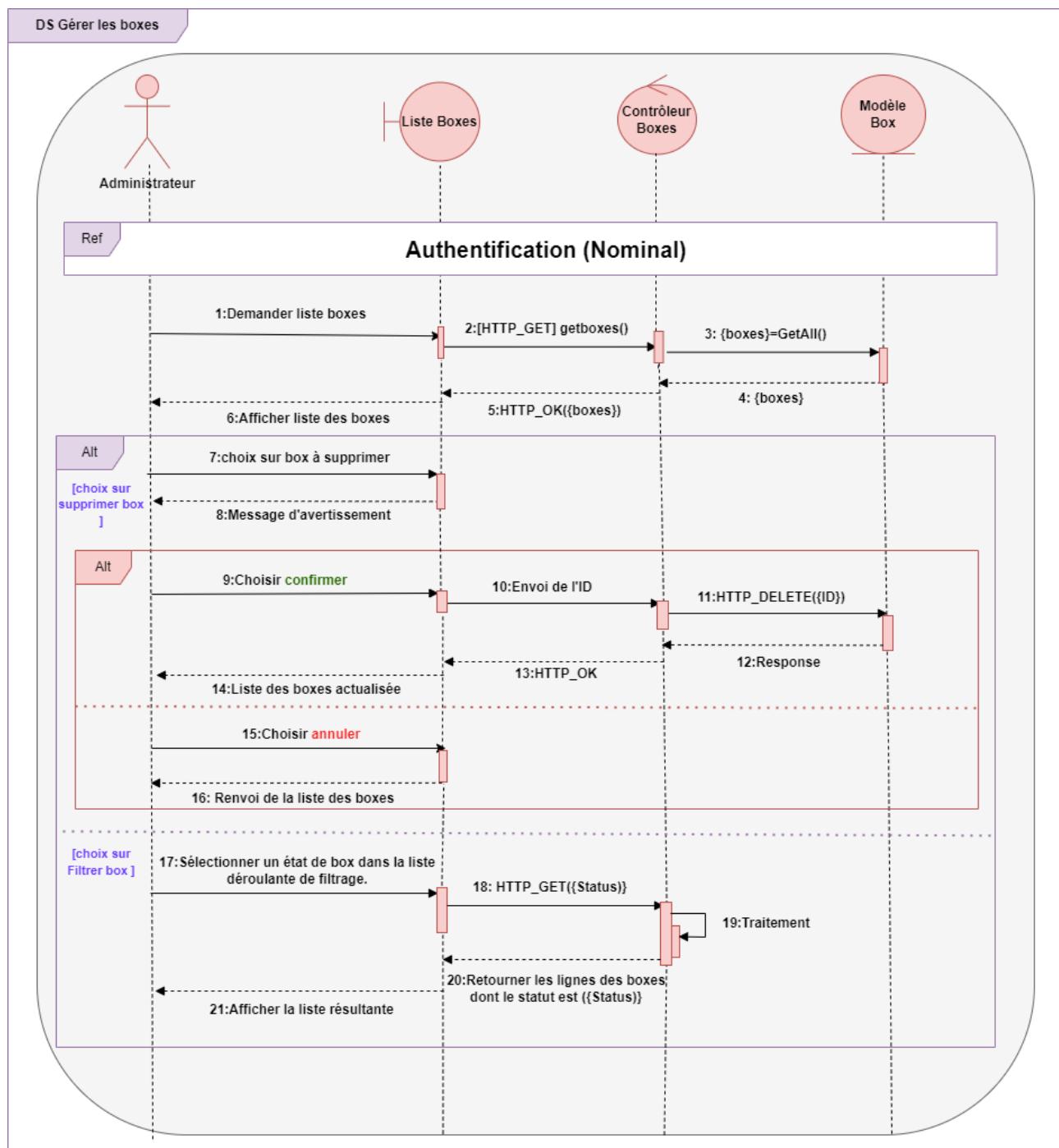


FIGURE 3.7 – DS du cas d'utilisation « Gérer les boxes »

3.2.3 Conception détaillée de sprint 2

3.2.3.1. Diagrammes de classes participantes

Le **diagramme de classes participantes (DCP)** est important puisqu'il effectue la jonction entre, d'une part, les cas d'utilisation, les modèles de la couche métiers et l'interface avec l'utilisateur. Il semble particulièrement important pour guider la phase de production du livrable final.

Les trois types de classes du DCP sont :

- **Dialogues** : Représentent les interfaces utilisateurs.
- **Entités** : Ne possèdent que les attributs représentants les informations persistantes de l'application.
- **Contrôles** : Intermédiaires entre les classes de dialogues et les classes d'entités. Elles ne possèdent que les opérations exprimant la logique d'application.

3.2.3.2. DCP du cas d'utilisation « S'authentifier »

Le DCP de ce cas d'utilisation s'authentifier est composé d'une classe de dialogue appelée "Connexion" qui communique avec le contrôleur "AuthController". Ce dernier vérifie auprès de l'entité "Utilisateur" si les informations saisies par l'utilisateur sont correctes ou non. Le contrôleur est également capable de récupérer le rôle de l'utilisateur qui vient de se connecter (Partenaire ou Client), son statut, ainsi que son identifiant, afin de les crypter dans le token d'authentification.

La figure 3.8 illustre le diagramme de classes participantes du cas d'utilisation « S'authentifier ».

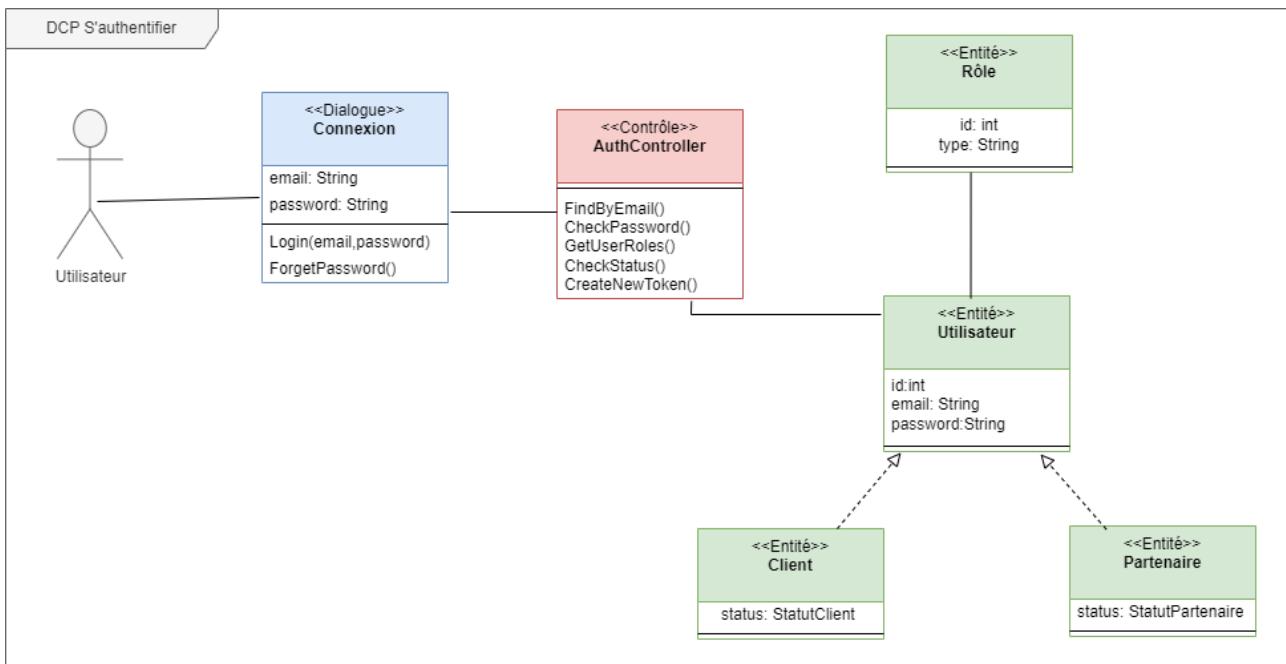


FIGURE 3.8 – DCP du cas d'utilisation « S'authentifier »

3.2.3.3. DCP du cas d'utilisation « Gérer profil »

Les informations concernant le partenaire connecté sont disponibles dans l'interface "Profil". Nous avons également une autre interface appelée "Statistiques" dans laquelle le partenaire peut consulter le nombre de commandes reçues et le nombre de ses boxes. En accédant à l'une de ces interfaces, le contrôle "PartnerController" chargera les détails du partenaire.

La figure 3.9 illustre le diagramme de classes participantes du cas d'utilisation « Gérer profil ».

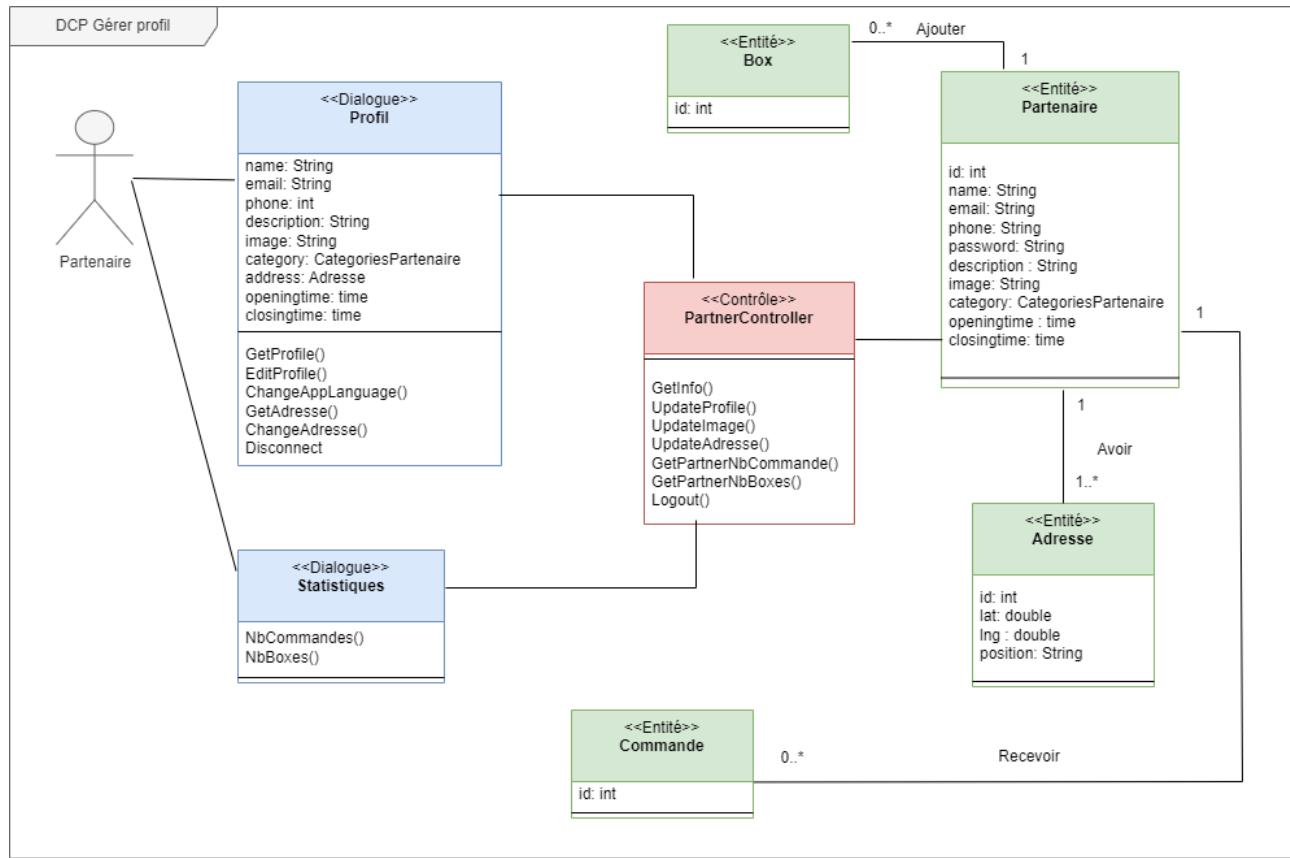


FIGURE 3.9 – DCP du cas d'utilisation « Gérer profil »

3.2.3.4. Cas d'utilisation « Gérer boxes »

1. DCP du cas d'utilisation « Gérer boxes »

Dans ce diagramme, nous trouvons six dialogues : un dialogue pour la création d'un nouveau box "NouveauBox" et les autres pour l'affichage et le suivi des boxes en fonction de leur état. Ces interfaces sont en relation avec le contrôle "BoxController", capable de réaliser les opérations appropriées et de manipuler l'entité nécessaire.

La figure 3.10 illustre le diagramme de classes participantes du cas d'utilisation « Gérer boxes ».

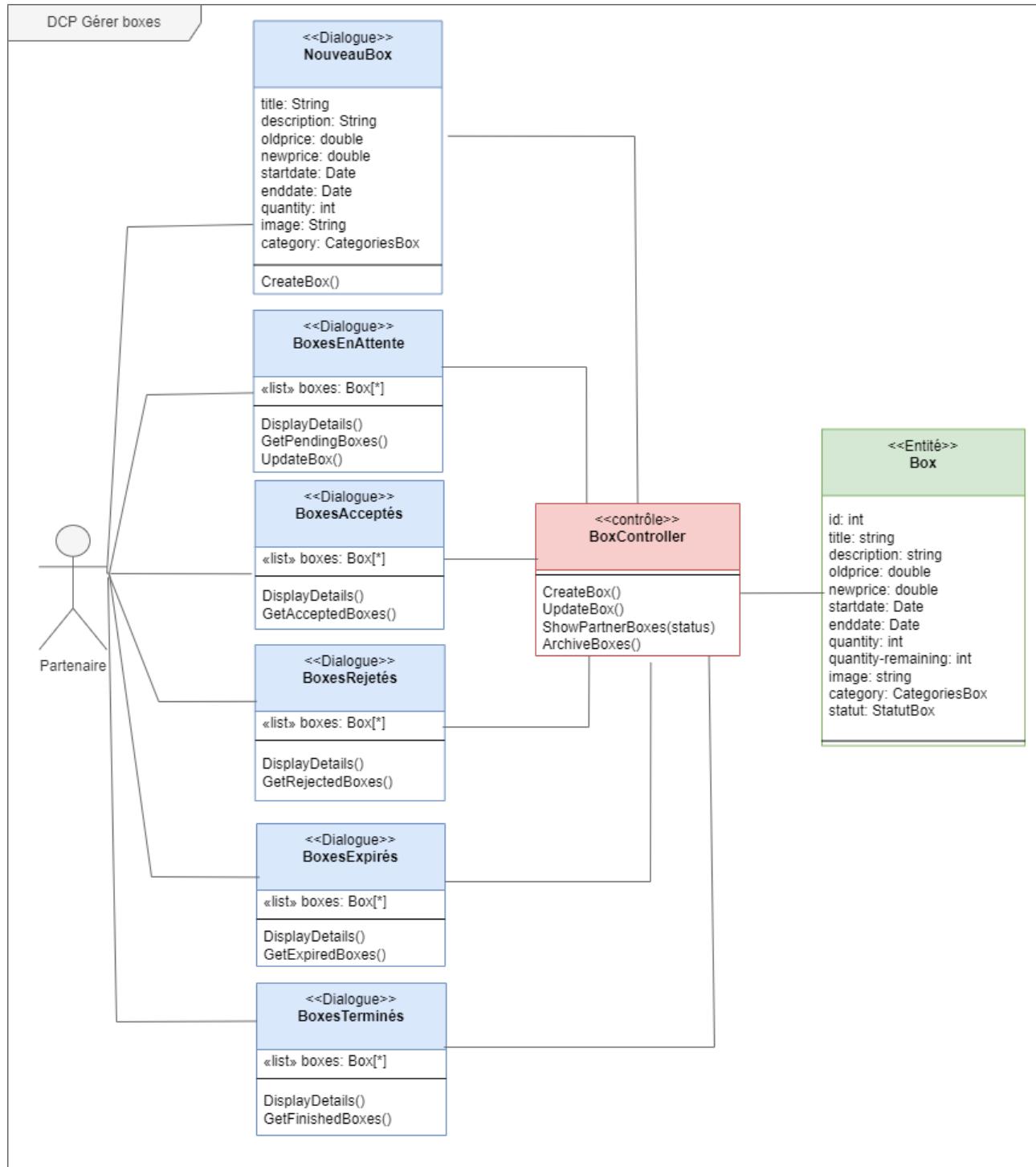


FIGURE 3.10 – DCP du cas d'utilisation « Gérer boxes »

2. Diagramme d'états-transitions du cas d'utilisation « Gérer boxes »

Un **diagramme d'états-transitions** est une représentation visuelle qui permet de décrire les états par lesquels une instance d'une classe peut **traverser**, et les transitions qui les lient. Chaque transition est **associée** un événement ou une action qui déclenche le passage à l'état suivant. **associée à**

passer

La figure 3.11 illustre le diagramme d'états-transitions du cas d'utilisation « Gérer boxes » et présente les différents états que la classe box peut traverser.

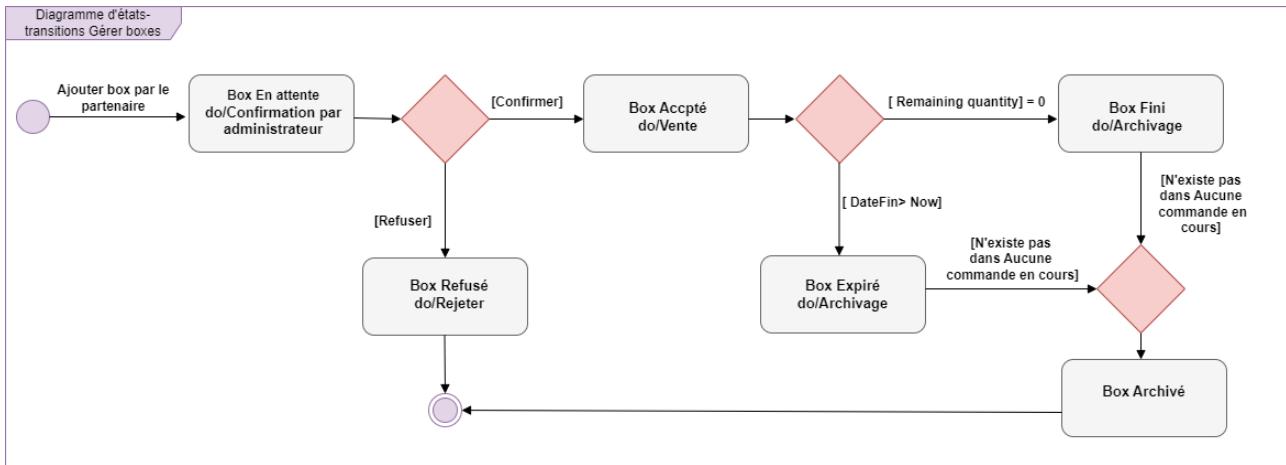


FIGURE 3.11 – Diagramme d'états-transitions du cas d'utilisation « Gérer boxes »

3.2.3.5. Cas d'utilisation « Consulter commandes »

1. DCP du cas d'utilisation « Consulter commandes »

Ce diagramme présente trois interfaces connectées à la classe de contrôle "CommandController", qui sera responsable de récupérer les données de toutes les commandes liées au partenaire connecté à partir des entités concernées.

La figure 3.12 illustre le diagramme de classes participantes du cas d'utilisation « Consulter commandes ».

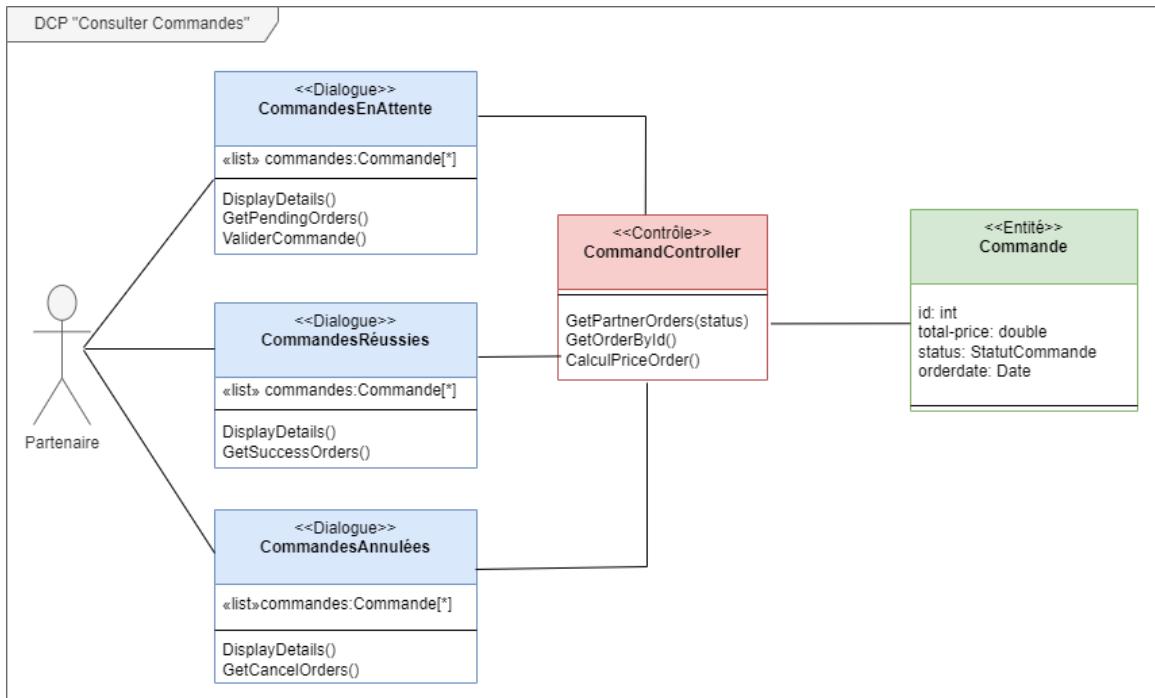


FIGURE 3.12 – DCP du cas d'utilisation « Consulter commandes »

2. DCC du cas d'utilisation « Consulter commandes »

Le **diagramme de classes de conception** présente en détail le séquencement d'**application**. **de l'application**

La figure 3.13 présente le diagramme de classes de conception détaillé autour de consulter commandes.

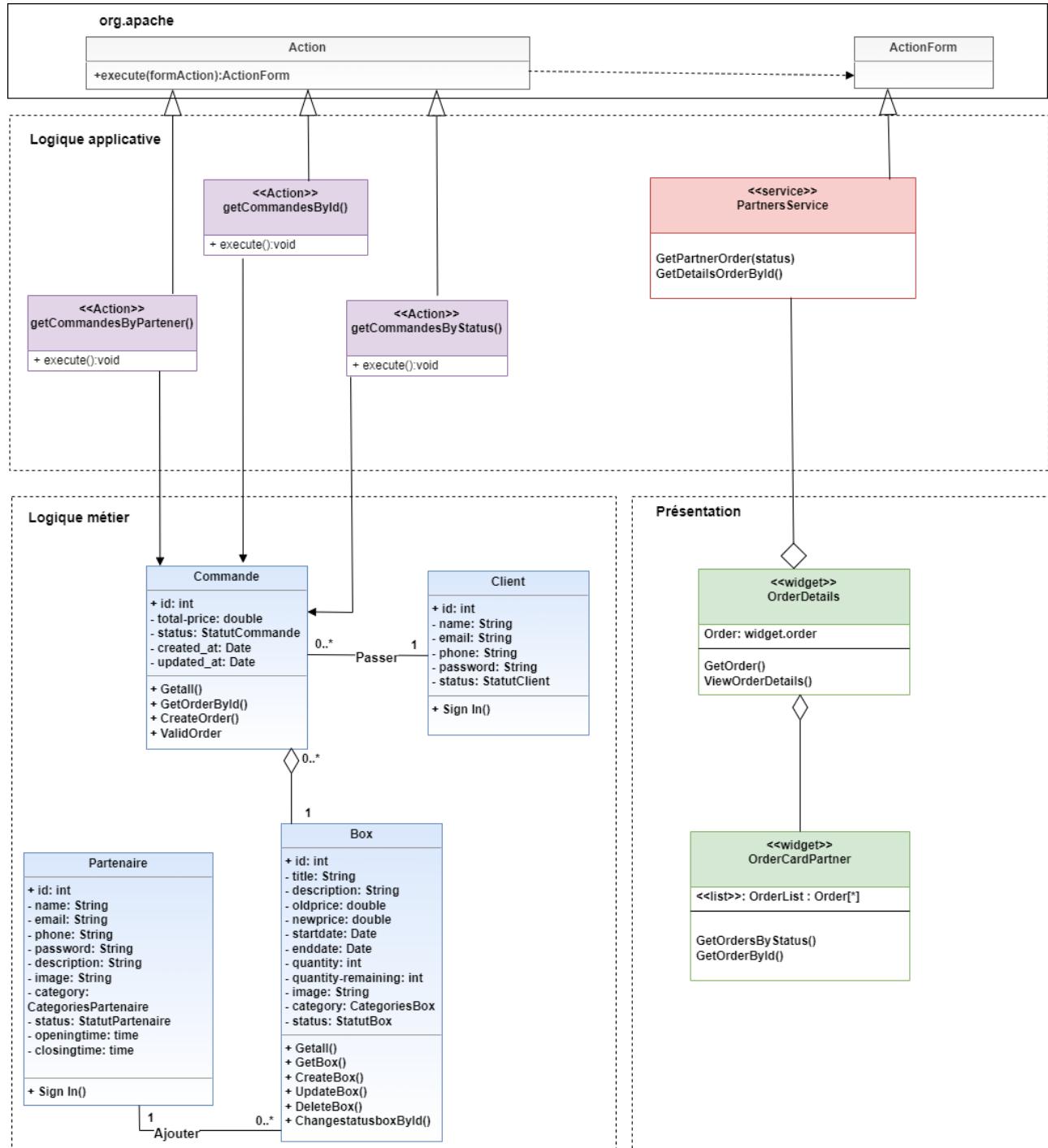


FIGURE 3.13 – DCC du cas d'utilisation « Consulter commandes »

3.2.4 Conception détaillée de sprint 3

3.2.4.1. DS du cas d'utilisation « Réserver Box »

La figure 3.14 illustre le diagramme de séquence du cas d'utilisation « Réserver Box » : Un client peut réserver un box.

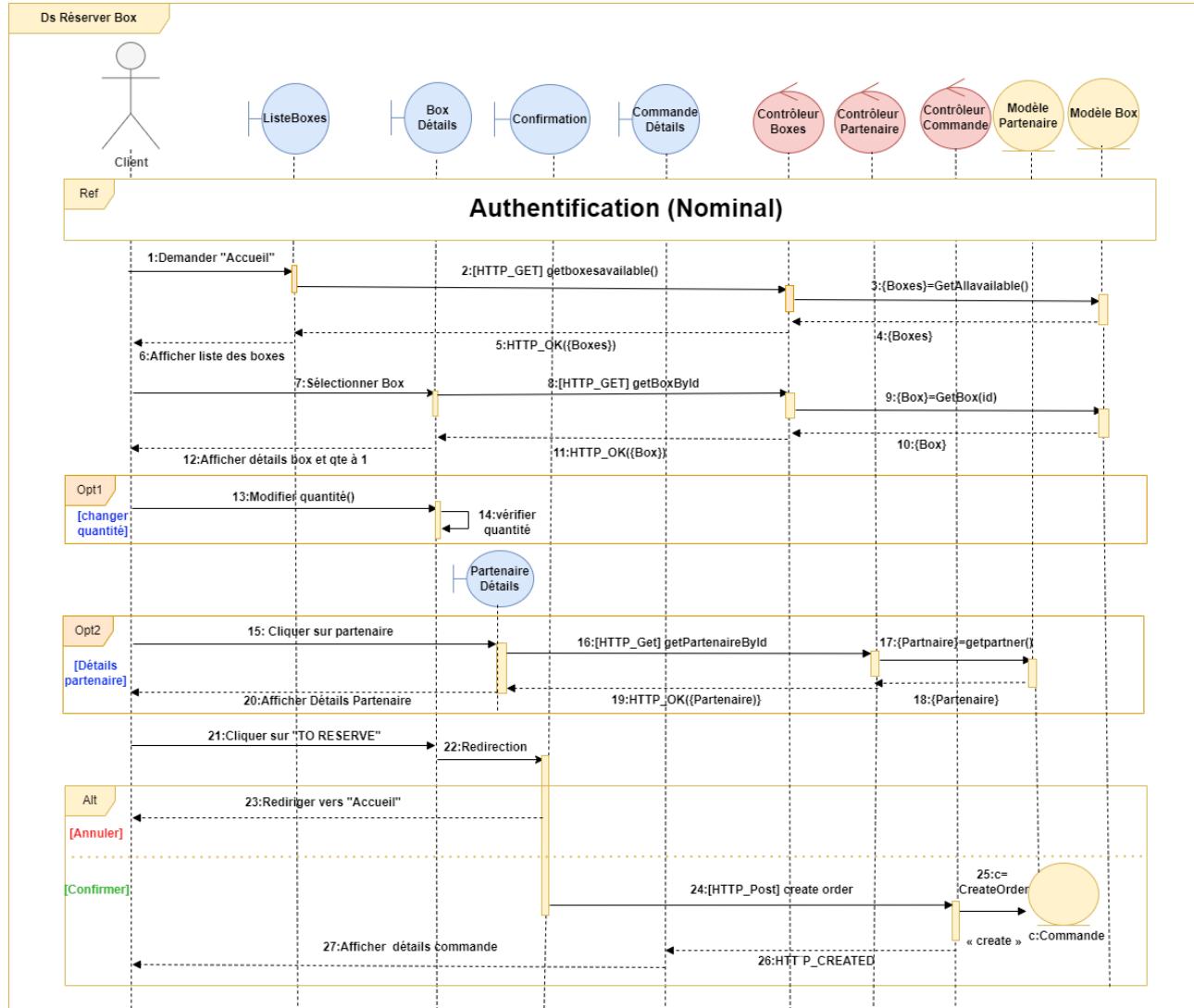


FIGURE 3.14 – DS du cas d'utilisation « Réserver Box »

3.2.4.2. DS du cas d'utilisation « Sélectionner boxes »

La figure 3.15 illustre le diagramme de séquence du cas d'utilisation « Sélectionner boxes » : Un client peut filtrer les boxes selon certains critères tels que le prix des boxes, les catégories de boxes et les catégories de partenaires.

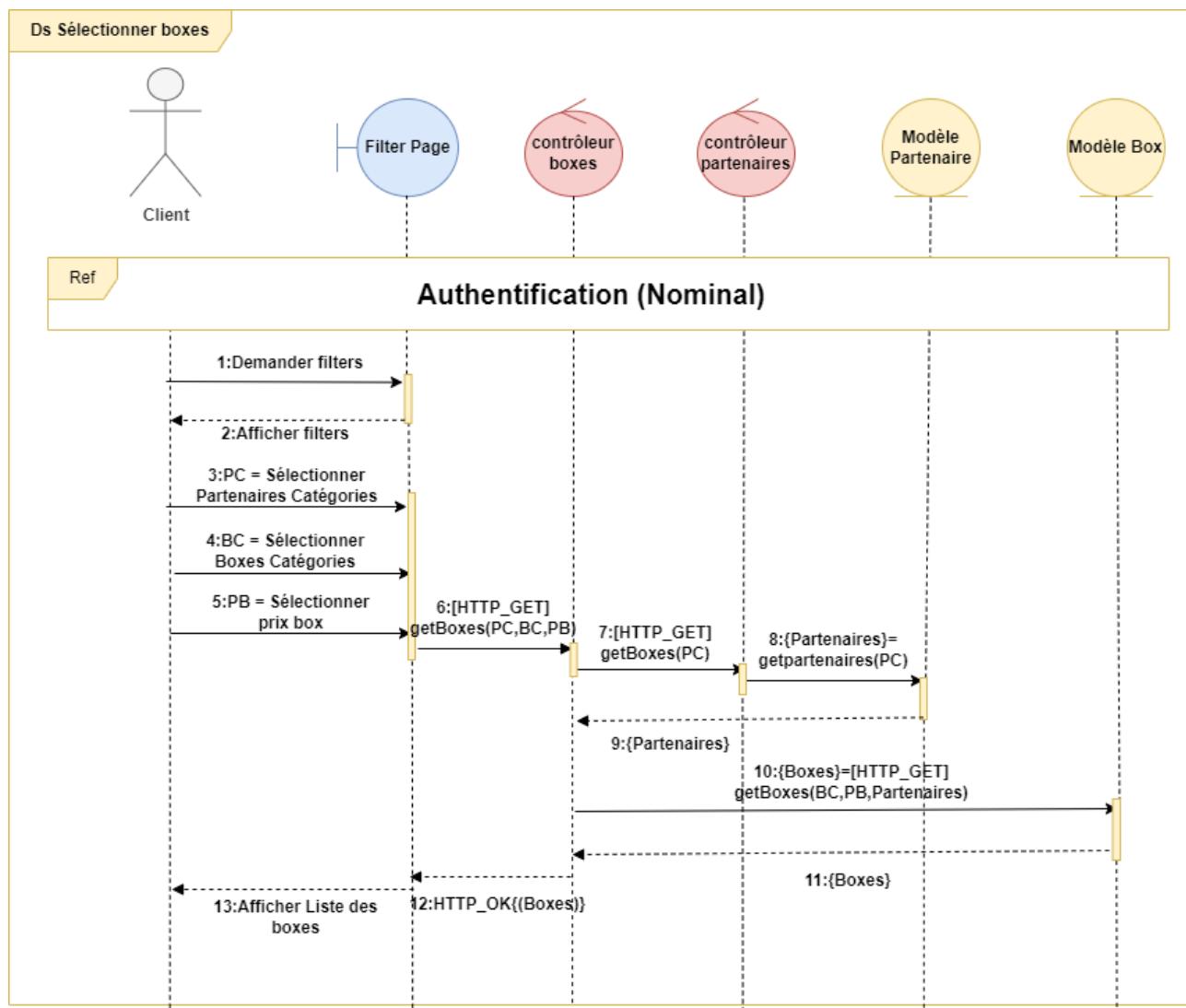


FIGURE 3.15 – DS du cas d'utilisation « Sélectionner boxes »

3.2.4.3. Diagramme d'activités

Le **diagramme d'activités** participe à la description du comportement d'un système et à sa modélisation dynamique. Nous avons utilisé ce diagramme pour le cas d'utilisation "Valider commandes", comme indiqué dans la figure 3.16.

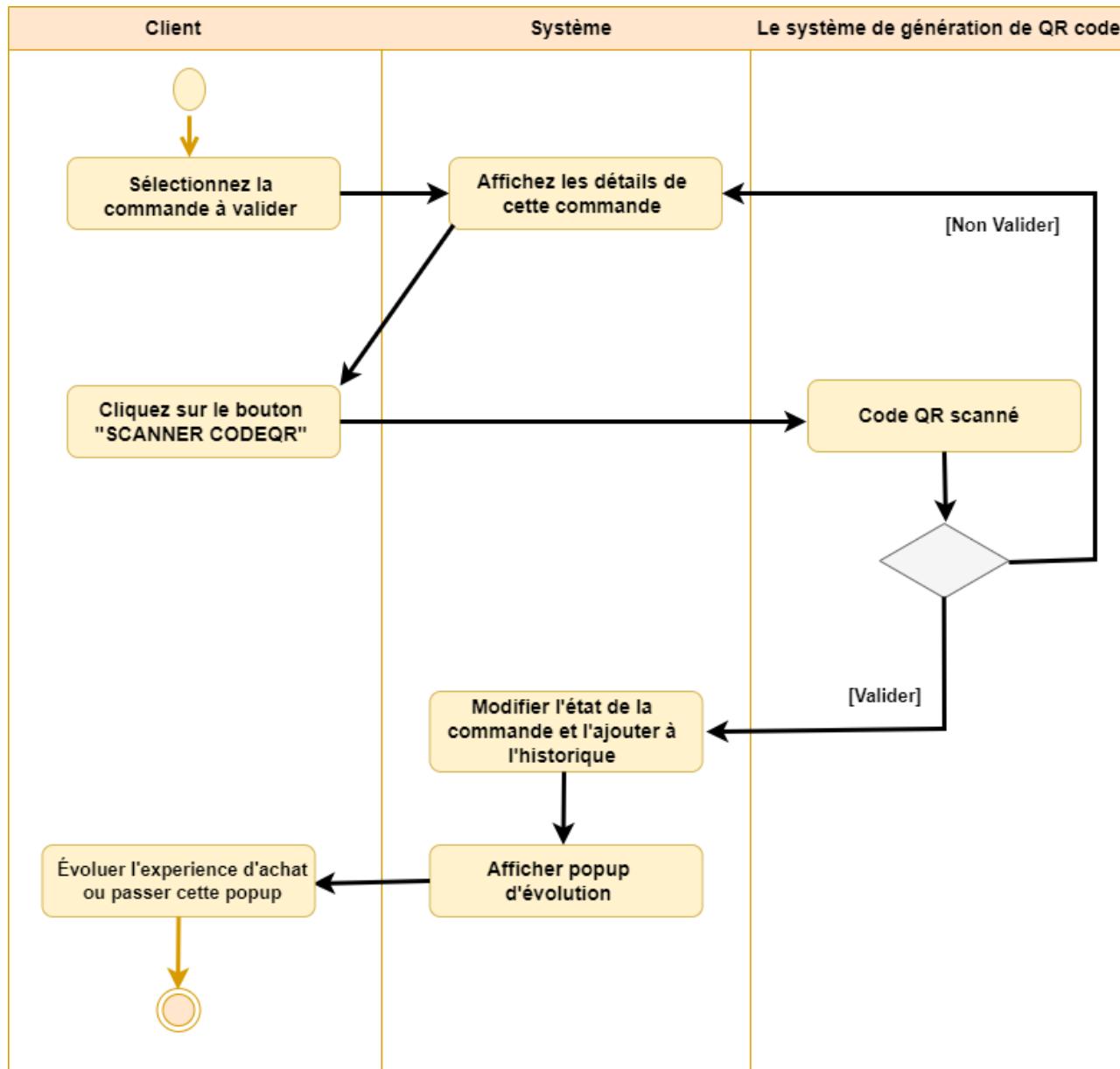


FIGURE 3.16 – DS du cas d'utilisation « Valider Commandes »

Conclusion

Au cours de ce chapitre, nous avons examiné le côté statique et le côté dynamique de notre projet, ainsi qu'une description approfondie des différents cas d'utilisation présents dans chaque sprint à travers plusieurs diagrammes et modèles UML. Dans ce qui suit, nous allons passer à la phase de réalisation pour mettre en œuvre notre application.

Chapitre 4

Réalisation

Introduction

Après avoir traité la partie conceptuelle, nous entamons la phase de réalisation de notre projet, au cours de laquelle nous présentons l'environnement logiciel et matériel dans lequel notre travail s'inscrit. Ensuite, nous exposons les différentes technologies utilisées pour la mise en place de notre système. Nous allons décrire également sa disposition physique à l'aide d'un diagramme de déploiement. Enfin, nous allons présenter les principales interfaces graphiques de l'application, illustrant les différentes fonctionnalités mises en place.

4.1 Spécification technique

Nous présentons dans cette section les choix techniques relatifs à l'environnement matériel et logiciel qui ont contribué à la réalisation de notre projet.

4.1.1 Environnement matériel

Au cours des différentes étapes de notre projet, à savoir la documentation, l'implémentation du code et les tests, nous avons disposé de :

Machine	Ordinateur Portable	Émulateur Virtuel
Marque	Lenovo	Pixel 3a API 30
Processeur	Intel Core I7	Android SDK platform 30(R)
RAM	8 Go	1536 MB
Système d'exploitation	Windows 10	android 11.0×86

TABLE 4.1 – Matériel de base

4.1.2 Environnement logiciel

Tout au long de la phase de développement, nous avons utilisé les outils logiciels et les langages de programmation suivant :

- **Éditeurs de développement :**

- **Visual Studio Code** : est un éditeur de code simplifié, qui est gratuit et développé par Microsoft. Il fonctionne sous Windows, mac OS et Linux. Il fournit aux développeurs à la fois un environnement de développement intégré avec des outils permettant de faire avancer les projets techniques, de l'édition, à la construction, jusqu'au débogage.[18]

Nous avons utilisé ce logiciel lors du développement de parties web et mobile.



Logo Visual Studio Code

- **Android Studio** : est un environnement de développement pour développer des applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle . Il peut être téléchargé sous les systèmes d'exploitation Windows, macOS, Chrome OS et Linux. [1]



Logo Android Studio

- **Langages de programmation :**

- **Dart** : est un langage optimisé pour le client permettant de développer des applications rapides sur n'importe quelle plate-forme. Son objectif est d'offrir le langage de programmation le plus productif pour le développement multiplateforme, associé à une plate-forme d'exécution flexible pour les frameworks d'applications. Dart constitué également la base de Flutter. Dart fournit le langage et les temps d'exécution qui alimentent les applications Flutter, mais Dart prend également en charge de nombreuses tâches de développement de base telles que le formatage, l'analyse et le test du code.[3]

Nous l'avons utilisé dans le développement de la partie mobile avec Flutter.



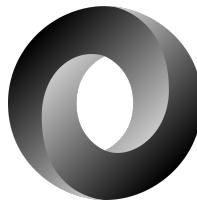
Logo Dart

- **PHP** : est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur web, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet. [14]
Nous l'avons utilisé également avec Laravel pour la partie backend.



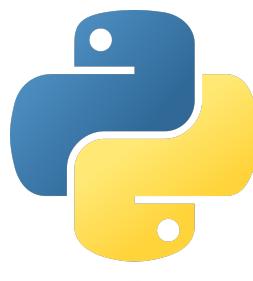
Logo PHP

- **JSON** : le JavaScript Object Notation (JSON) est un format standard utilisé pour représenter des données structurées de façon semblable aux objets Javascript. [8]
Dans notre projet il nous a servi lors de l'échange de données entre notre back et le front de notre application facilitant ainsi une communication fluide entre les deux parties.



Logo JSON

- **Python** : est un langage de programmation interprété, multiparadigme et multiplate-formes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions. [16]
Nous avons utilisé pour la réalisation du module intelligent.



Logo Python

- **TypeScript** : est un langage de programmation libre développée par Microsoft qui a pour but d'améliorer et de sécuriser la production de code JavaScript. Il s'agit d'un sur-ensemble syntaxique strict de JavaScript (c'est-à-dire que tout code JavaScript

correct peut être utilisé avec TypeScript). Le code TypeScript est transcompilé en JavaScript, et peut ainsi être interprété par n'importe quel navigateur web ou moteur JavaScript. [17]

Nous l'avons également utilisé avec Angular pour la partie frontend de notre partie web.



Logo TypeScript

- **Système de gestion de base de données :**

- **MySQL** : un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde.[12]



Logo MySQL

- **Neo4j** : est un système de gestion de bases de données orienté graphe. À la différence des systèmes classiques, son approche n'est pas fondée sur l'algèbre relationnelle mais sur la théorie des graphes. Les données sont stockées de manière assez libre (sans modèle prédéterminé) dans des nœuds reliés entre eux par des relations (des arcs porteurs d'une sémantique forte). Neo4j utilise un langage d'interrogation spécifique CYPHER, conçu pour être assez intuitif.[13]



Logo Neo4j

- **Structure de développement :**

- **Flutter** : est le framework de Google permettant un développement multiplateforme. Il permet donc de ne développer une application qu'une seule fois pour les différentes plateformes : iOS et Android. Flutter s'appuie sur le langage de programmation DART, créé également par Google et présenté au public en 2011.[6]
Nous l'avons également utilisé pour la partie frontend de notre partie mobile.



Logo Flutter

- **Laravel** : est un framework PHP multi-plateforme permettant de créer des applications web. Laravel est principalement un framework de développement Backend, bien qu'il offre quelques fonctionnalités Frontend. Cependant, de nombreuses fonctionnalités de Laravel sont agnostiques au niveau du Frontend. [10]
Nous l'avons utilisé pour la partie backend de notre partie web et mobile.



Logo Laravel

- **Angular** : est un framework JavaScript développé par Google. Il est utilisé pour développer des applications web et mobile. Avec cette technologie, on réalise des interfaces de type monopage qui fonctionnent sans recharge de la page web.[2]
Nous avons employé cet outil pour développer l'infrastructure frontend de notre partie web.



Logo Angular

- **Flask** : est un micro framework de développement web en Python. Il est classé comme microframework car il est très léger. Flask a pour objectif de garder un noyau simple mais extensible.[5]
Nous avons employé cet outil pour développer l'API du module intelligent.



Logo Flask

- **Outils informatiques :**

- **POSTMAN** : sert à exécuter des appels HTTP directement depuis une interface graphique. Vous pourrez simplement choisir l'URL, la méthode HTTP (le plus souvent GET, POST, PUT, PATCH et DELETE), les headers, les query params et dans certains cas le body de la requête.[15]
Nous avons l'utiliser comme un outil de test d'API qui nous permet de tester et de déboguer notre API plus facilement.



Logo POSTMAN

- **Git** : est un logiciel de gestion de versions qui suit l'évolution des fichiers sources et garde les anciennes versions de chacun d'eux sans rien écraser. Cela permet de retrouver les différentes versions d'un fichier ou d'un lot de fichiers connexes et ainsi éviter des problèmes.[7]



Logo Git

- **Draw.io** : est un outil de création de diagrammes en ligne. Il permet aux utilisateurs de concevoir et de dessiner facilement divers types de diagrammes tels que des diagrammes de flux, des diagrammes UML, et bien d'autres encore.



Logo Draw.io

- **Outil de rédaction du rapport :**

- **Overleaf** : est une plateforme en ligne permettant de rédiger et d'éditer des documents LaTeX de manière collaborative et en temps réel.



Logo Overleaf

4.2 Diagramme de déploiement

Le **diagramme de déploiement**, inclus dans la catégorie des diagrammes structurels, sert à décrire l'architecture physique des ressources matérielles constituant le système. Il permet de mettre en évidence les liens entre les différents composants du système et de visualiser la disposition des nœuds.

Quant à notre cas, l'utilisateur a la possibilité d'utiliser la partie web ou mobile. Les deux parties sont connectées au même API, qui communique ensuite avec le serveur de la base de données. La figure 4.1 illustre l'architecture physique de notre application.

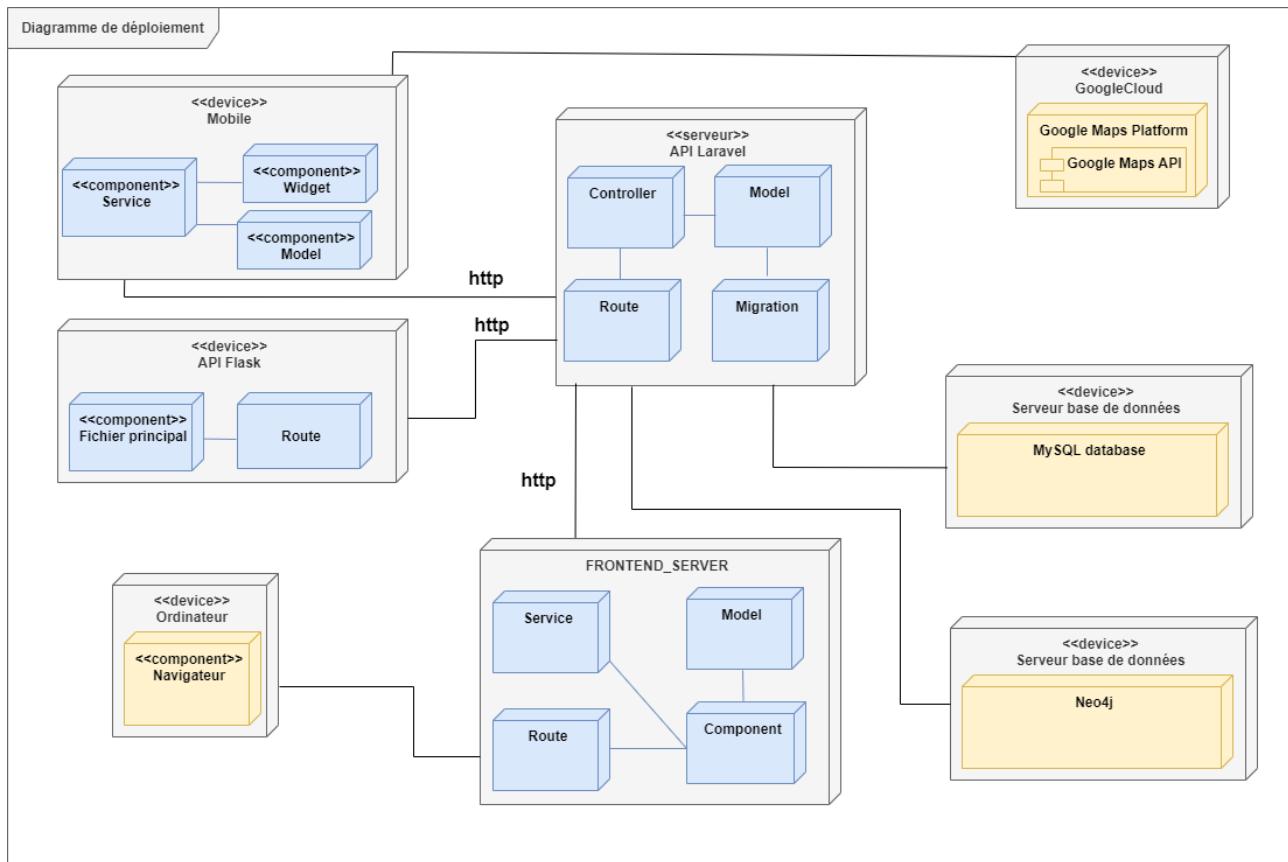


FIGURE 4.1 – Diagramme de déploiement

4.3 Réalisation

Dans cette section, nous nous intéressons à présenter la partie de la réalisation du projet. Nous présentons les scénarios d'exécution en donnant un aperçu sur la réalisation des fonctionnalités ainsi que les principales interfaces de notre application dans chaque sprint.

4.3.1 Réalisation du sprint 1 : Interfaces des administrateurs

4.3.1.1 Authentification pour l'administrateur

Avant chaque accès à la partie web, chaque administrateur doit impérativement saisir son email et son mot de passe via l'interface d'authentification. L'application vérifie

également si les informations saisies correspondent à celles d'un administrateur existant dans la base de données. La figure 4.2 illustre l'interface en question.

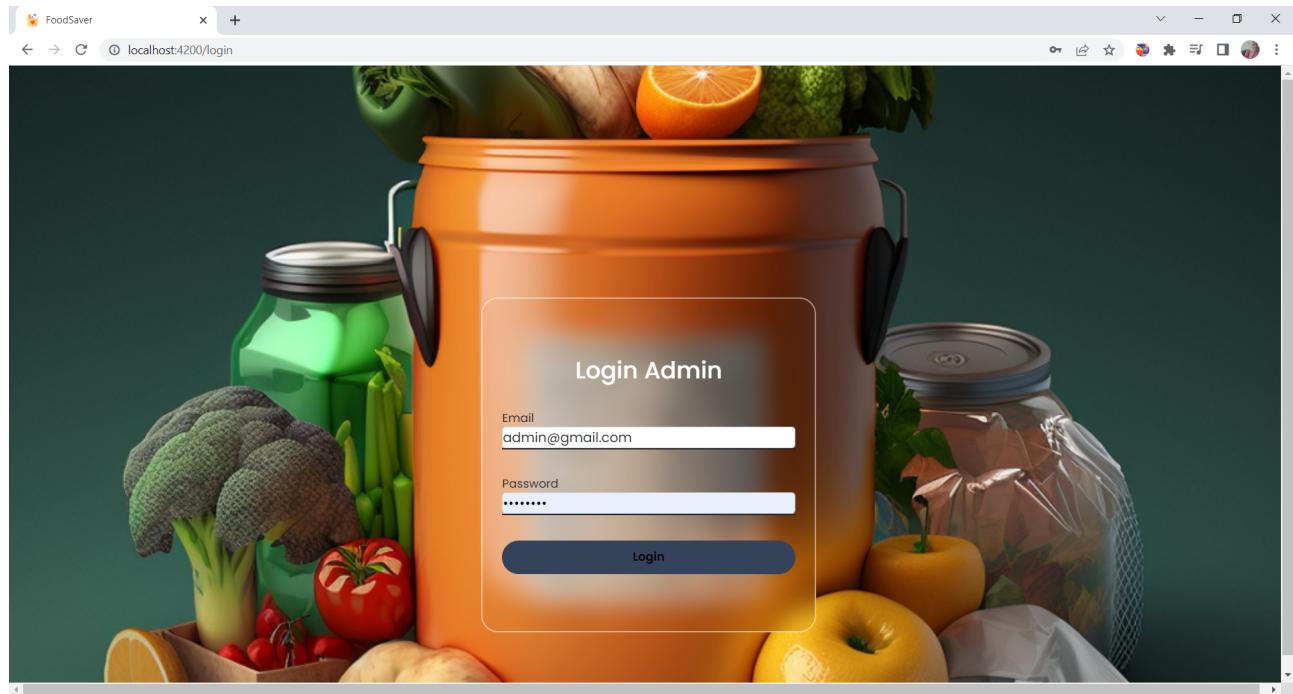
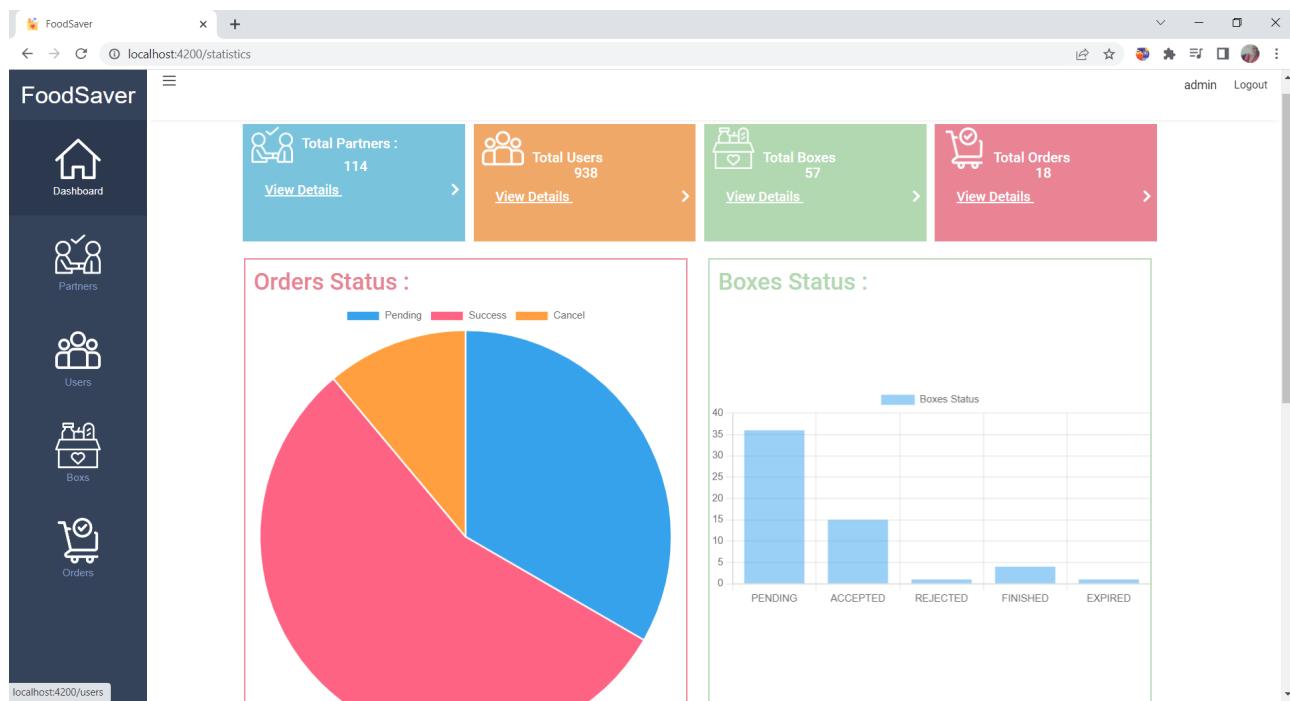


FIGURE 4.2 – L'interface d'authentification pour l'administrateur

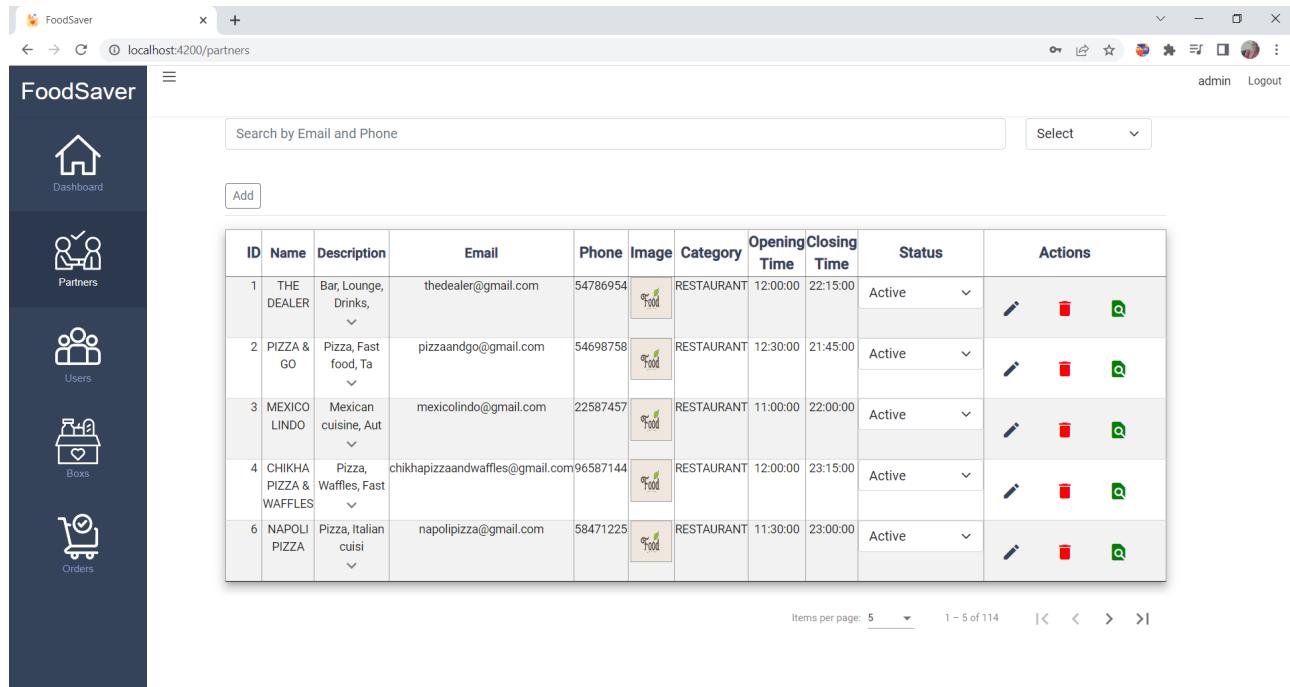
4.3.1.2. Tableau de bord

Une fois l'authentification effectuée, l'administrateur aura accès à une interface qui contient des statistiques relatives à son rôle, telles que décrites dans la figure 4.3. Ces statistiques concernent notamment les partenaires, les clients, les commandes et les boxes.

**FIGURE 4.3 – L'interface de tableau de bord**

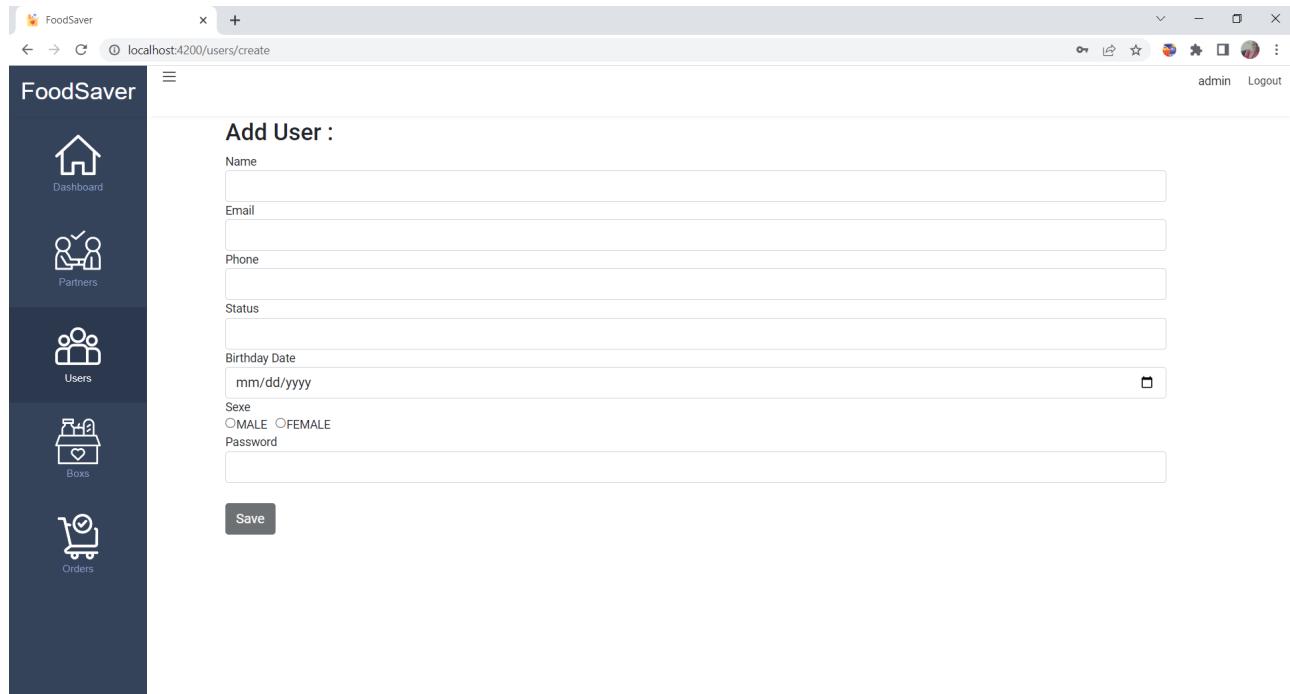
4.3.1.3. Gestion des partenaire

Notre application offre à l'administrateur la rubrique "Partners" permettant de gérer l'ensemble des partenaires de la partie mobile. En effet, il peut créer, modifier, supprimer, chercher, modifier status et filtrer un partenaire, ainsi que lister les partenaires existants et consulter les informations détaillées d'un partenaire en particulier. La figure 4.4 illustre l' interface de gestion des partenaires.

**FIGURE 4.4 – L'interface de gestion des partenaires**

4.3.1.4. Gestion des clients

La section "Users" offre à l'administrateur une gamme complète de fonctionnalités de gestion des clients de la partie mobile. L'administrateur peut facilement créer, chercher, modifier, supprimer, activer, désactiver et filtrer des clients, ainsi que consulter la liste complète des clients existants et accéder aux informations détaillées d'un client en particulier. La figure 4.5 présente l'interface dédiée à l'ajout d'un client.



The screenshot shows a web browser window for the FoodSaver application. The title bar reads "FoodSaver" and the address bar shows "localhost:4200/users/create". The left sidebar has a dark blue background with five icons: "Dashboard" (house), "Partners" (two people), "Users" (three people), "Boxes" (box with heart), and "Orders" (shopping cart). The main content area is titled "Add User :". It contains several input fields: "Name" (text input), "Email" (text input), "Phone" (text input), "Status" (text input), "Birthday Date" (text input with placeholder "mm/dd/yyyy" and a calendar icon), "Sexe" (radio buttons for "MALE" and "FEMALE"), and "Password" (text input). A "Save" button is at the bottom.

FIGURE 4.5 – L'interface d'ajout de client

4.3.1.5. Gestion des boxes

La section "Boxes" propose un ensemble complet de fonctionnalités pour gérer les boxes. L'administrateur peut créer, modifier, supprimer, chercher, changer le statut et filtrer les boxes. De plus, il peut consulter la liste complète des boxes existantes et accéder aux informations spécifiques à chaque partenaire de box. Pour avoir une meilleure idée de ces fonctionnalités, la figure 4.6 montre l'interface dédiée à la gestion des boxes.

ID	Title	Description	Old Price	New Price	Start Availability	End Availability	Quantity	Remaining Quantity	Image	Category	Status	Actions
16	mexico	mexican food and juice	20.00	15.00	2023-06-22	2023-06-23	3	1		PREPARED DISHES	ACCEPTED	
17	box antiwaste	pizza,coca,pizza quatre saison	30.00	22.00	2023-06-22	2023-06-23	2	1		PREPARED DISHES	ACCEPTED	
18	box antiwaste	2kg of meat	70.00	45.00	2023-06-22	2023-06-23	3	1		MEAT	ACCEPTED	
19	freshjuice	1,5L juice,crepe,sweets	25.00	15.00	2023-06-22	2023-06-23	4	2		DRINKS	ACCEPTED	
20	box markt	food market,fresh produce	30.00	15.00	2023-06-22	2023-06-23	5	1		FRUITS AND VEGETABLES	ACCEPTED	

FIGURE 4.6 – L’interfaces de gestion des boxes

4.3.1.6. Gestion des commandes

La section "Orders" fournit un ensemble exhaustif de fonctionnalités pour la gestion des commandes. L'administrateur dispose d'outils conviviaux pour créer, chercher, supprimer et filtrer les commandes. De plus, il peut consulter la liste complète des commandes existantes et accéder aux informations détaillées d'une commande spécifique. La figure 4.7 illustre l'interface qui présente les détails d'une commande.

#	Box Title	Quantity	Price	Partner Id	Partner Name	Partner Email	Partner Phone	Partner Category
17	box antiwaste	1	22.00	2	PIZZA & GO	pizzaandgo@gmail.com	54698758	RESTAURANT

FIGURE 4.7 – L’interface de détails d’une commande

4.3.2 Réalisation du sprint 2 : Interfaces des partenaires

4.3.2.1. Interface « Autorisation »

Lorsque vous lancez l'application pour la première fois, une fenêtre d'autorisation apparaît afin de vous permettre d'accorder l'accès à votre GPS comme montre la figure 4.8.



FIGURE 4.8 – L'interface d'autorisation

4.3.2.2. Interfaces « Accueil »

Après avoir lancé l'application et terminé le téléchargement, la page d'accueil s'affiche. À partir de cette page, l'utilisateur a la possibilité de choisir entre s'inscrire en tant que partenaire ou en tant que client, ou bien de s'authentifier, comme indiqué dans la figure 4.9.

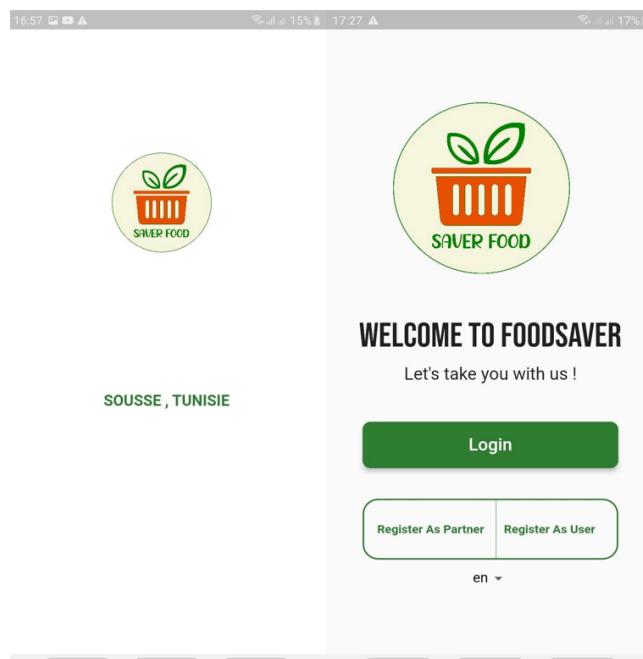


FIGURE 4.9 – Les interfaces d'accueil

4.3.2.3. Interfaces « Inscription »

Pour que le partenaire puisse accéder au service de notre application, il doit d'abord s'inscrire. Après l'inscription, le partenaire doit attendre la validation de l'administrateur pour que son compte soit activé comme indique la figure 4.10.

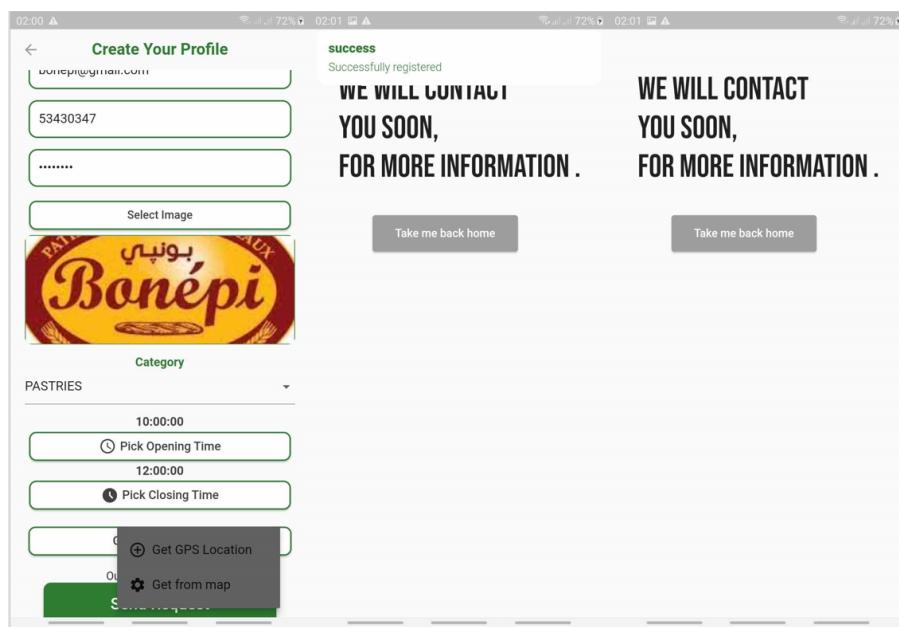


FIGURE 4.10 – Les interfaces d'inscription de partenaire

4.3.2.4. Interface « Authentification »

L'authentification est une étape essentielle pour accéder aux fonctionnalités de notre application. Cette interface permet à l'utilisateur de confirmer son identité en fournissant

une adresse e-mail et un mot de passe valides. En interagissant avec notre base de données, elle vérifie la validité du compte et détermine le rôle de l'utilisateur, qui peut être un partenaire ou un client.

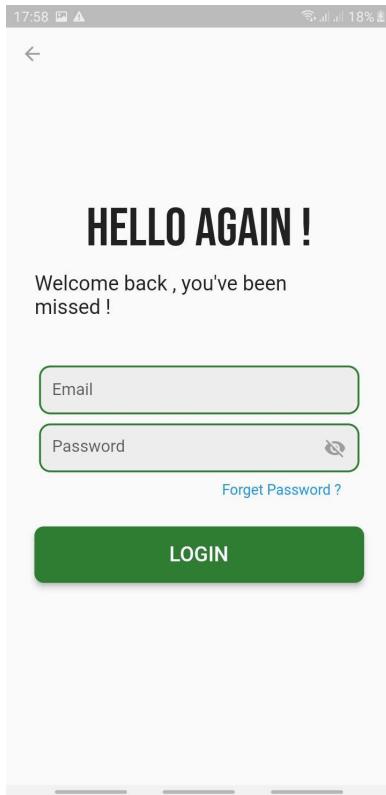


FIGURE 4.11 – L'interface d'authentification

4.3.2.5. Interfaces « Gérer les commandes »

Dès que le partenaire est connecté, il a la possibilité de gérer ses commandes en les consultant en fonction de leur état. Il peut afficher les détails de chaque commande et valider celles en attente en scannant le code QR fourni par le client, comme illustré dans la figure 4.12.

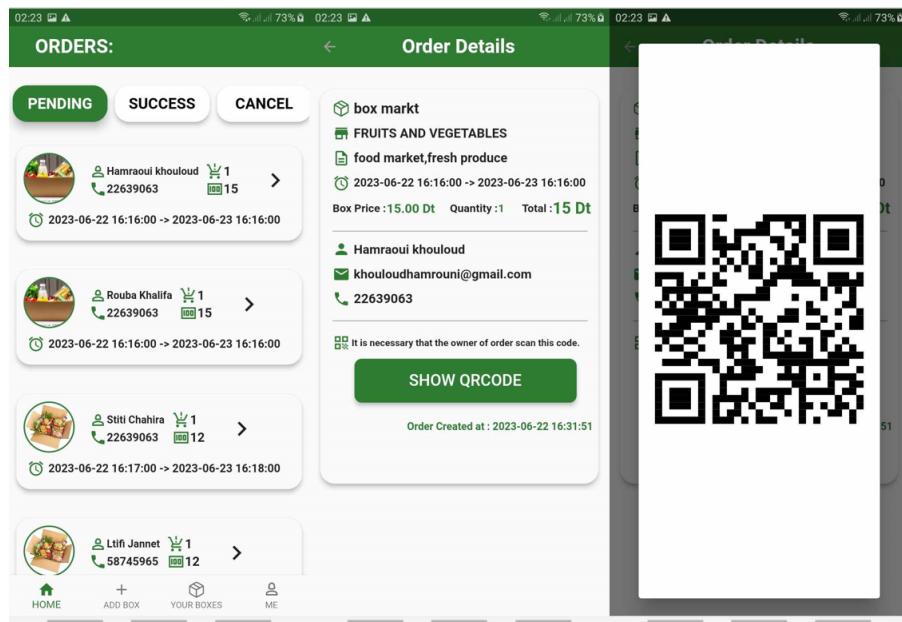


FIGURE 4.12 – Les interfaces de gestion des commandes

4.3.2.6. Interfaces « Gérer les boxes »

Le partenaire peut gérer ses boxes en les consultant en fonction de leur état. Il a également le droit d'ajouter de nouveaux box et de modifier les boxes en attente, comme illustré dans la figure 4.13.

nouveaux boxes

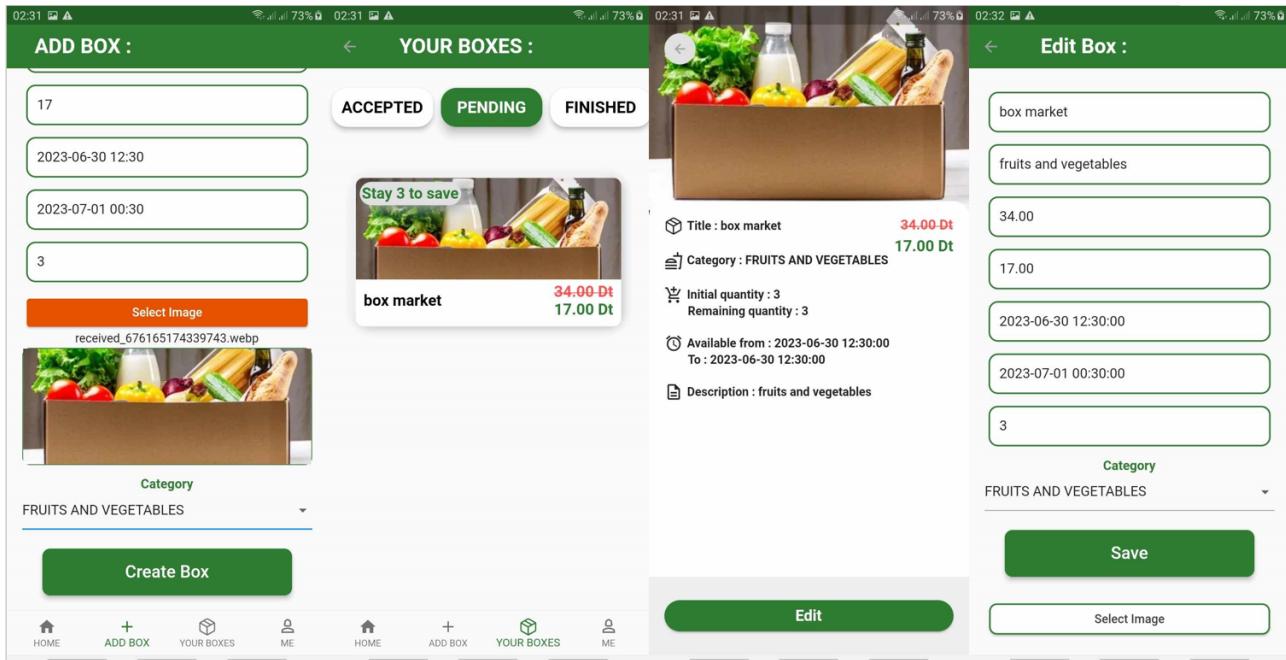


FIGURE 4.13 – Les interfaces de gestion des boxes

4.3.2.7. Interfaces « Gérer son profil »

Le partenaire a la possibilité de gérer son profil en consultant et en modifiant ses informations, en modifiant son adresse et en consultant ses statistiques comme l'indique la figure 4.14.

l'indique

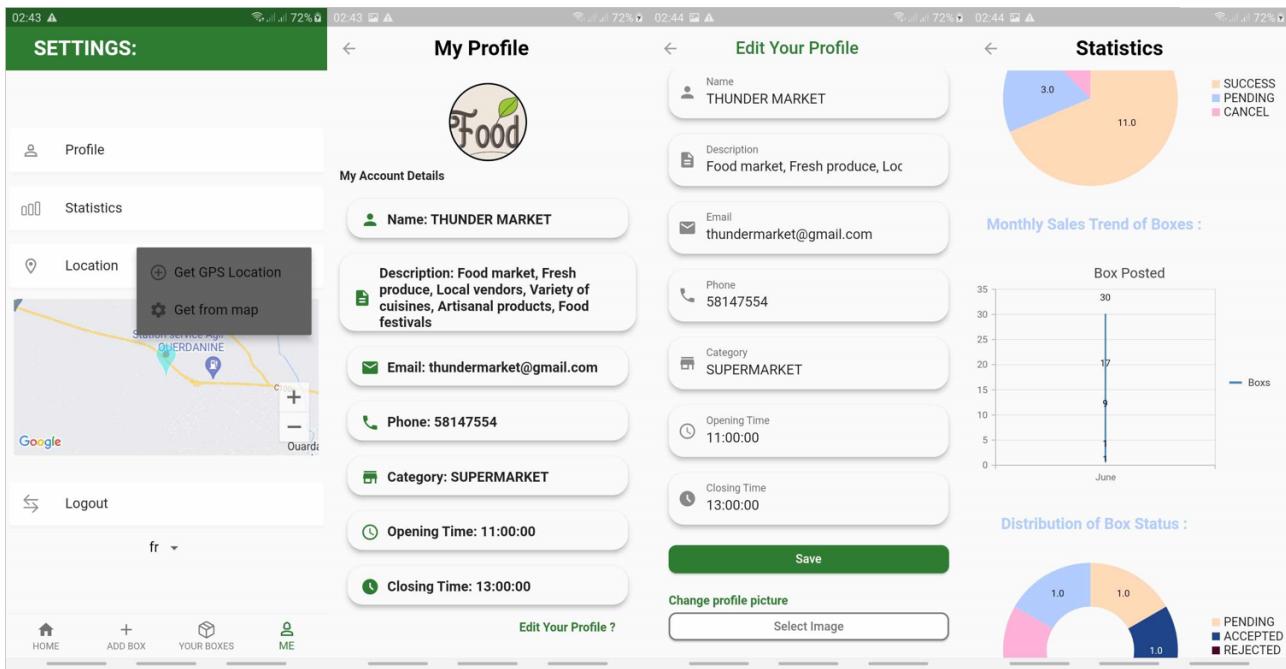


FIGURE 4.14 – Les interfaces de gestion de profil

4.3.3 Réalisation du sprint 3 : Interfaces des clients

4.3.3.1. Interfaces « Oublié mot de passe »

Quel que soit l'utilisateur (client ou partenaire) ayant oublié son mot de passe, il doit cliquer sur "Mot de passe oublié" pour ouvrir une nouvelle interface. Ensuite, il renseigne son adresse e-mail pour recevoir un code par e-mail. Une fois l'adresse e-mail vérifiée, l'utilisateur peut modifier son mot de passe et accéder à son interface, comme illustré dans la figure 4.15.

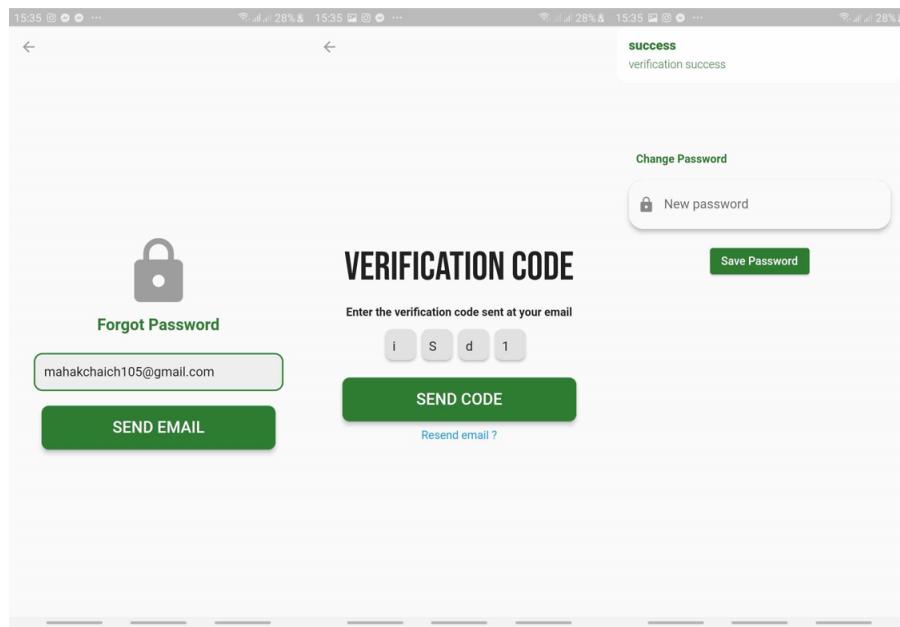


FIGURE 4.15 – Les interfaces de récupération de mot de passe

4.3.3.2. Interfaces « Consulter Accueil »

Le client a la possibilité de consulter son accueil qui contient les boxes disponibles et d'accéder à leur description. De plus, en consultant les détails de chaque box, il peut également accéder aux informations concernant le partenaire associé à ce box comme illustré dans la figure 4.16.

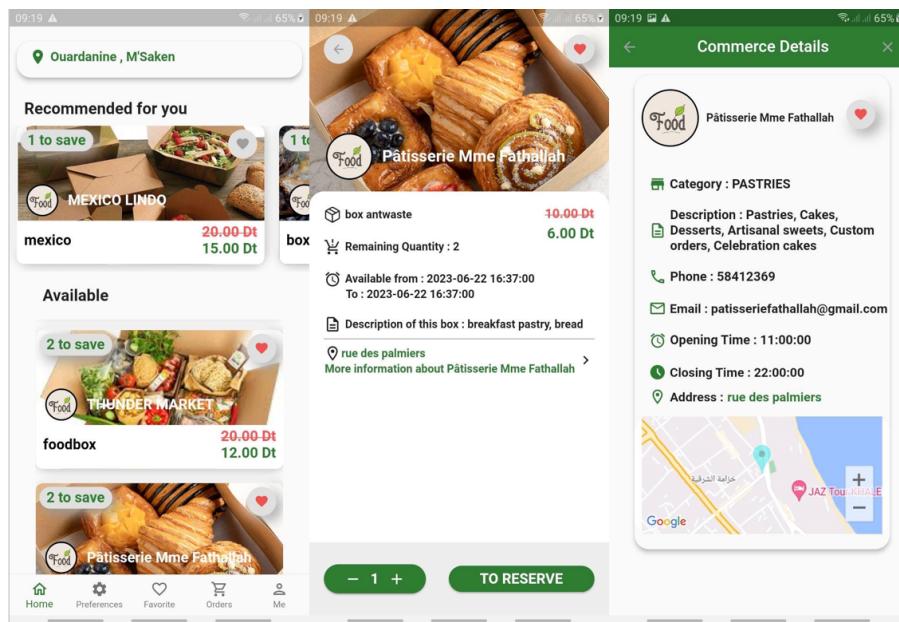


FIGURE 4.16 – Les interfaces de consultation d'accueil

4.3.3.3. Interfaces « Passer commande »

Le client a la possibilité de modifier la quantité de box et de passer une commande. Une interface de confirmation s'affiche, et il lui suffit de confirmer sa commande pour afficher les détails de celle-ci, comme illustré dans la figure 4.17.

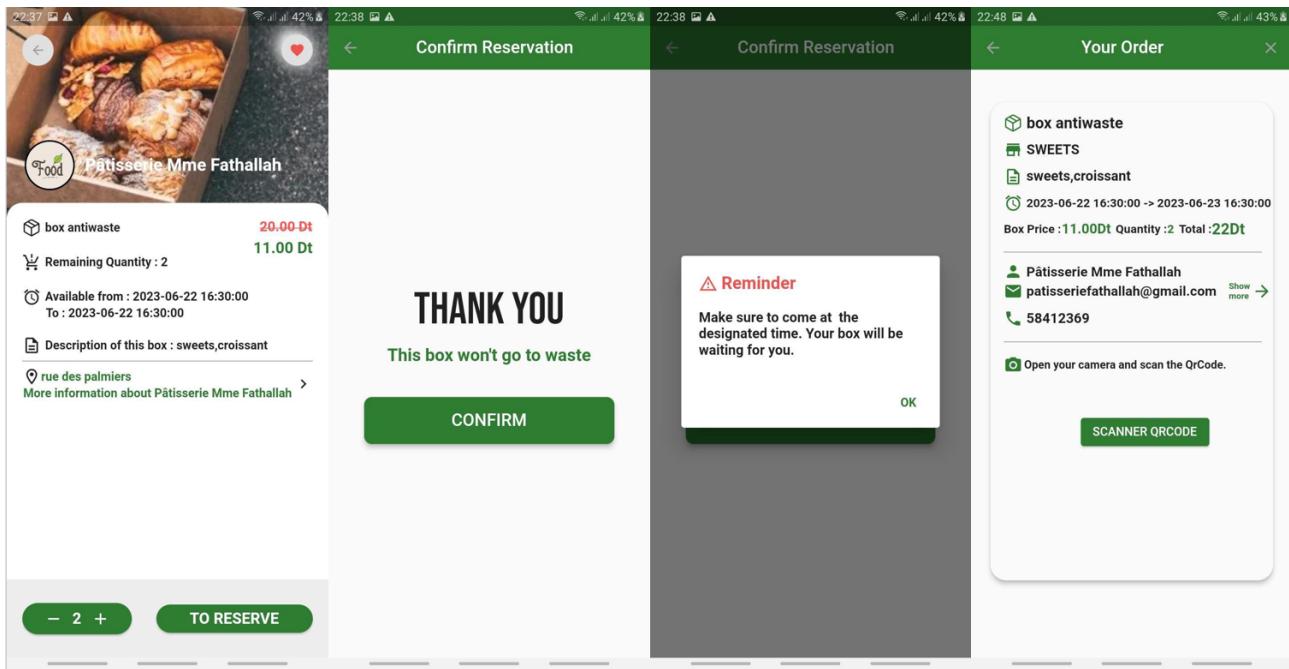


FIGURE 4.17 – Les interfaces pour passer commande

4.3.3.4. Interfaces « confirmer commande »

Lors de la récupération de la commande, le client scanne le code QR pour confirmer qu'il récupère sa commande. Cela déplace la commande vers l'historique des commandes et affiche une interface d'évaluation permettant au client de donner son avis sur l'expérience d'achat.

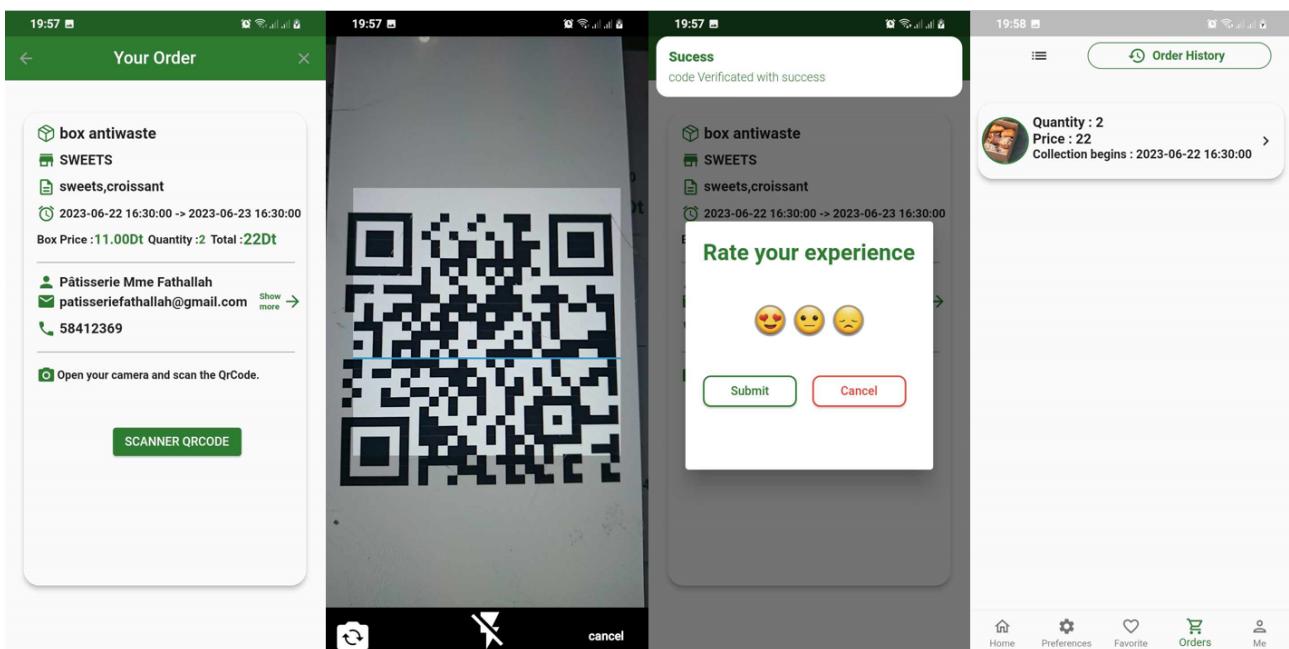


FIGURE 4.18 – Les interfaces de confirmation de commande

4.3.3.5. Interfaces « Sélectionner boxes »

Le client a la possibilité d'appliquer des filtres pour trier les boxes selon certains critères,

tels que le prix, les catégories des boxes et les catégories des partenaires. De plus, le client peut spécifier une distance afin de visualiser les partenaires situés à proximité de lui, comme indiqué sur la figure 4.19.

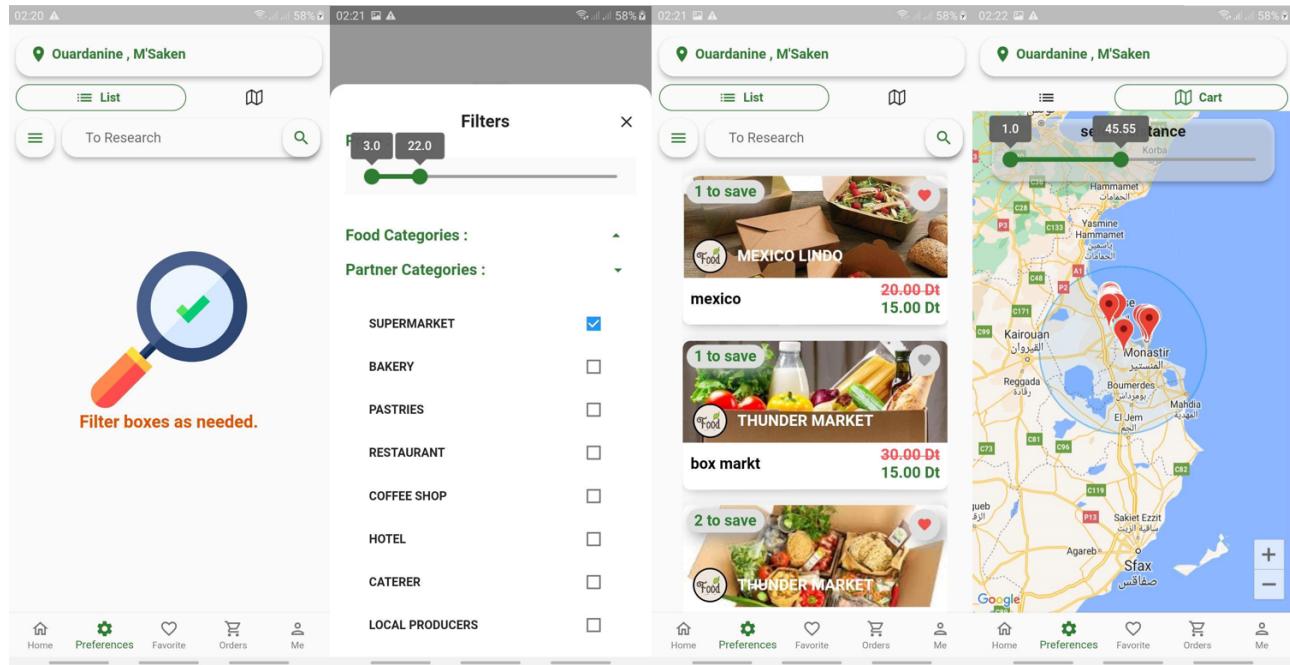


FIGURE 4.19 – Les interfaces de sélection de boxes

4.3.3.6. Interfaces « Consulter les favoris »

Le client a la possibilité d'inclure des partenaires et des boxes dans sa liste de favoris, et il peut également y accéder pour les consulter, comme indiqué sur la figure 4.20.

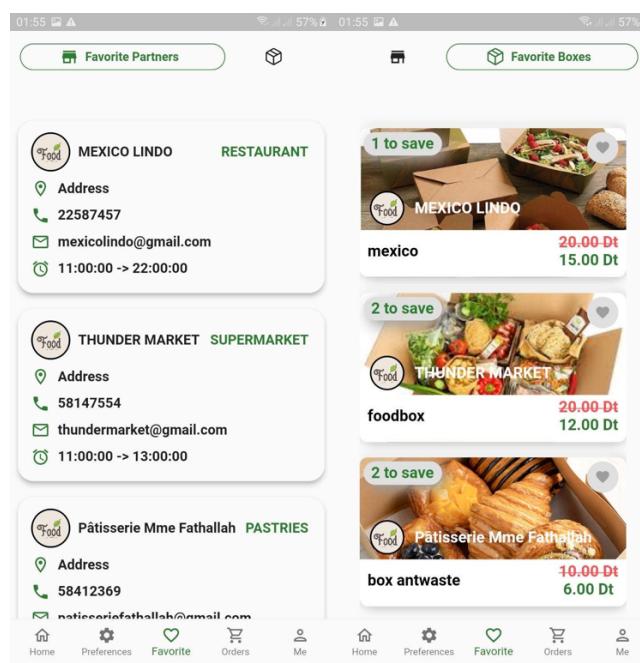


FIGURE 4.20 – Les interfaces de consultation des favoris

4.3.3.7. Interfaces « Gérer son profil »

qu'il a commandé

Le client a la possibilité de visualiser le nombre de boxes qu'il achète, de connaître le montant d'argent qu'il gagne, ainsi que de consulter et de modifier ses informations personnelles dans son profil.

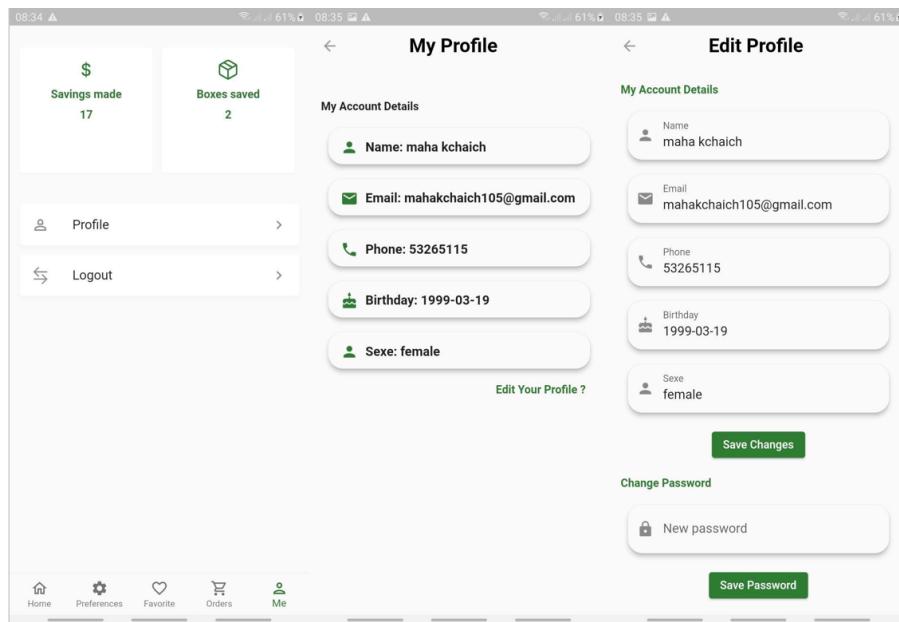


FIGURE 4.21 – Les interfaces de gestion de profil

4.3.4 Réalisation du sprint 4

Dans le but d'améliorer nos ventes et d'offrir une expérience utilisateur personnalisée, nous avons développé un système de recommandation intelligente. Ce système nous permet de répondre aux besoins spécifiques de chaque client en lui suggérant des boxes qui correspondent à ses préférences.

En utilisant l'intelligence artificielle, notre système analyse les interactions des clients sur notre plateforme. Ensuite, il génère des recommandations personnalisées basées sur ces données, afin d'aider nos clients à découvrir de nouveaux boxes qui pourraient les intéresser.

4.3.4.1. Méthodologie

Puisque notre système est basé sur les réactions des clients, nous avons décidé de mettre en place deux systèmes : un système basé sur l'intelligence artificielle (IA) et un autre système créé avec Neo4j spécifiquement pour les nouveaux clients.

1. Construction du système basé sur l'intelligence artificielle (IA) :

La figure 4.31 présente le flux de travail que nous devons suivre pour obtenir notre système de recommandation.

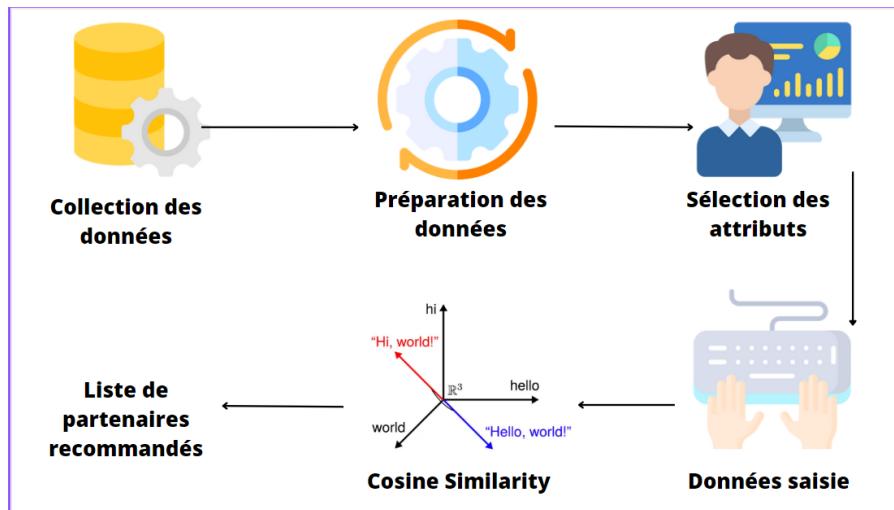


FIGURE 4.22 – Flux de travail

Nous avons pris la décision de nous concentrer sur la recommandation des partenaires plutôt que des boxes, étant donné que notre application n'a pas encore été déployée ni utilisée par le public. Par conséquent, nous ne disposons pas encore de données sur les ventes de boxes, mais nous avons des informations sur les partenaires. Dans cette optique, nous choisissons de recommander les partenaires et, à partir de cette recommandation, nous suggérons les boxes proposées par ces partenaires. Cette approche est temporaire, en attendant que notre application soit disponible au public et que nous puissions commencer à collecter des données pertinentes.

- **Construction de jeu de données :**

Dans notre projet, nous souhaitons que nos résultats attendus reflètent au mieux la réalité. Étant donné que Webify prévoit de fusionner les deux applications, SaverFood et Thunder Express, l'application de livraison de repas à domicile, nous avons décidé d'anticiper cette fusion en demandant des données réelles provenant de Thunder Express. Cela nous permettra de réaliser notre système de manière plus précise.

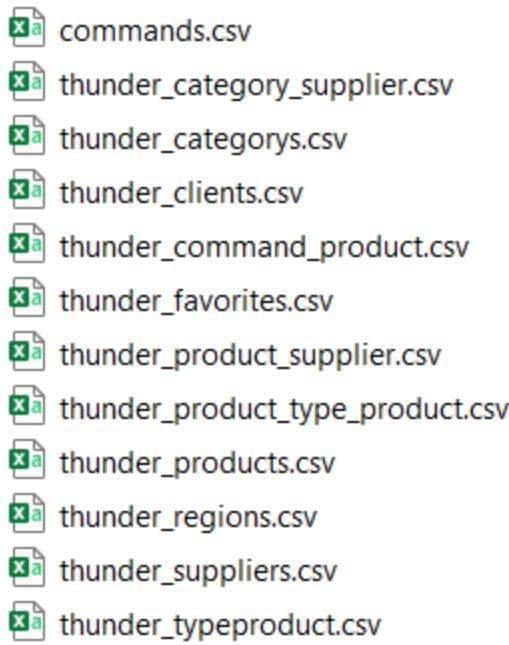


FIGURE 4.23 – Données de l'application en ligne ThunderExpress

• Préparation des données :

- **Nettoyage des données et sélection des caractéristiques** : étant donné que notre ensemble de données ne présente ni valeurs nulles ni entrées redondantes, nous avons procédé à la sélection des caractéristiques en supprimant les colonnes qui ne sont pas utiles pour recommander des partenaires.

thunder_suppliers.csv - Excel													
A	B	C	D	E	F	G	H	J	K	L	M	N	
1	id	name	description	email	phone	image	adress		category	openingtime	closingtime		
2	1 THE DEALER	Bar, Lounge, Drinks, Nightlife, Craft cocktails, Live @ thedealer@gmail.com	54780954	received_J Rue Youssef Saheb Tabia - prét rond point Mashmoun RESTAURANT						12:00	22:3		
3	2 PIZZA & GO	Pizza, Fast food, Takeaway, Quick bites, Delivery, A pizzaandgo@gmail.com	54698758	received_J Cité Nozha Kantaoui Hmme Sousse à coté La Village RESTAURANT						12:30	21:4		
4	3 MEXICO LINDO	Mexican cuisine, Authentic flavors, Tacos, Burritos, mexicocindo@gmail.com	22587457	received_J Rue 14 Janvier - en face Ooredoo avant rond point K RESTAURANT						11:00	22:0		
5	4 CHIKHA PIZZA & WAFFLES	Pizza, Waffles, Fast food, Desserts, Family-friendly, chikhappizzawaffles@gmail.com	96587144	received_J Avenue des oranges - à coté Café Route 66 - Hmme RESTAURANT						12:00	23:1		
6	6 NAPOLI PIZZA	Pizza, Italian cuisine, Pasta, Authentic flavors, Thin-napolipizza@gmail.com	58471225	received_J Rue Yasser Arafat - à coté Carré Noir - Sahloul - Sour RESTAURANT						11:30	23:0		
7	7 BOCCONCINI	Italian cuisine, Cheese, Fine dining, Pasta, Wine sele bocconcini@gmail.com	44100199	received_J Rue Yasser Arafat - Immeuble Hasdrubal - Sahloul - RESTAURANT						11:00	22:4		
8	8 BRUSCHETTA	Italian cuisine, Bruschetta, Appetizers, Wine bar, Ot bruschetta@gmail.com	54542001	received_J Rue Yasser Arafat - à coté Carré Noir - Sahloul - RESTAURANT						11:30	22:1		
9	10 MY 3RD ARGOS	Grill, Fast food, Tacos, Mexican cuisine, Gyros, My3rdargos@gmail.com	54132088	received_J Rue Dr Chabeb Hmme Moslem - en face Zaitouna bank - Sahloul - RESTAURANT						12:00	21:3		
10	11 TRATTORIA DA FRANCO	Italian cuisine, Pasta, Traditional dishes, Family fri trattoriafranco@gmail.com	29070710	received_J Rue Sadok Mkaadem - en face Dar El Kunafa - Kzammart RESTAURANT						12:30	21:0		
11	12 PIZZERIA BIANCO	Italian cuisine, Wood-fired oven, Neapolitan pizzeriabianco@gmail.com	45874669	received_J Taher sfer - Sahloul - RESTAURANT						2:43	14:4		
12	13 FRESH JUICE & CRÊPES	Juice bar, Smoothies, Crepes, Healthy options, Bresfreshjuiceandcrepes@gmail.com	41258963	received_J Rue Sadok Laroussi - près Café Don Pablo - Sahlol - RESTAURANT						12:00	20:0		
13	14 Papillon pizza by chef souße	Pizza, Gourmet, Chef's specials, Upscale dining, Wlpapillonpizza@yahoo.fr@sousse@gmail.com	96548777	received_J 2 avenue des oranges						2:50	14:5		
14	15 kfc	Fast food, Fried chicken, Burgers, Drive-thru, Comb kfc@gmail.com	26987456	received_J Taher sfer						15:20	16:2		
15	21 TANDOOR LA CARTE	Indian cuisine, Tandoori dishes, Curries, A la carte tandoorlacarte@gmail.com	45879125	received_J Rue Hassan Ibn Said - près Café Montecello - Khemis RESTAURANT						12:00	20:3		
16	22 MR FOOD	Fast Food, Burgers, Sandwiches, Quick bites, Value imrfood@gmail.com	5747987	received_J Rue Charles De Gaulle - près Dar Chabeb Hmme Sour RESTAURANT						11:00	23:3		
17	23 LE MEXICAIN TACOS ET BURRITOS	Mexican cuisine, Tacos, Burritos, Spicy flavors, Freslemexicaintacosteburritos@gmail.com	42115987	received_J Rue Jalloul Ben Cherifa - près Café Black Box - Souss RESTAURANT						11:30	23:0		
18	24 K-FOOD	Korean cuisine, Bibimbap, Kimchi, BBQ, Banchan (si_k_food@gmail.com)	25100506	received_J Rue Sadok Mkaadem - à coté Café White House - Sahlol - RESTAURANT						12:00	22:0		
19	25 CHEDEN FOOD	Local cuisine, Traditional dishes, Authentic flavors, chedenfood@gmail.com	26102103	received_J Rue Imem Moslem - près Dar Kunaifa - Kzammart RESTAURANT						11:00	23:0		
20	26 BANDE PIZZA	Pizza, Artistic theme, Traditional dishes, Creative toppings, bandepizza@gmail.com	45740333	received_J Rue Dr Matougui - à coté Café Montecello - Khemis RESTAURANT						12:00	23:1		
21	27 KARDO FOOD	Tunisian cuisine, Traditional dishes, Spices, Traditional fit karodofit@gmail.com	25324240	received_J Rue Dr Matougui - à coté Café Montecello - Khemis RESTAURANT						11:30	0:0		
22	28 PARADAM KHEMMA	Tunisian cuisine, Bathbab sandwiches, Mediterani mybatbab@gmail.com	3247265	received_J Rue 14 Janvier - Route touristique - à coté Galeria - RESTAURANT						12:00	23:3		
23	30 MY BATBOUT	Moroccan cuisine, Bathbab sandwiches, Mediterani mybatbab@gmail.com	32698775	received_J Rue Abdellahim Ben Badis - près Café Tropic - LOCAL PRODUCERS						11:30	23:3		
24	31 PARADAM SAHLOUL	Tunisian cuisine, Local dishes, Spices, Traditional fit paradamsahoul@gmail.com	27485996	received_J Rue Boukhris - en face Poste Sahoul - Sousse RESTAURANT						11:30	23:3		
25	32 THE MOE'S DINER	American cuisine, Burgers, Milkshakes, Traditional fit paradamsahoul@gmail.com	54125877	received_J Rue Dr Maghreb Arabe - près rond point Msadeek M RESTAURANT						11:30	22:3		
26	34 LA MAISON DES VIANDES	Steakhouse, Grill, Meat dishes, Fine dining, Premium lamaisondesviandes@gmail.com	42587114	received_J Rue Abdellahid Daghir - en face Baladiat Sahoul GF LOCAL PRODUCERS						12:00	20:0		
27	36 ANBAR	Lebanese cuisine, Middle Eastern flavors, Mezze, Fi.anbar@gmail.com	52698774	received_J Rue les palmiers - à coté Café Chelli - Sousse COFFEE SHOP						11:30	21:0		
28	37 COCKTAIL FOOD	Cocktails, Small plates, Tapas, Bar food, Mixology, cocktailfood@gmail.com	4598774	received_J Rue Ittihad - près rond point Mosaïque - Sahloul - RESTAURANT						11:15	21:3		
29	38 DAR AL AILA	Arabic cuisine, Traditional dishes, Tea, Oriental swe daralala@gmail.com	25147896	received_J Rue Saad Bin Oubada - près L'instant Café - Sousse LOCAL PRODUCERS						0:00	23:0		

FIGURE 4.24 – Données initial

Nous avons donc décidé de supprimer l'email, le numéro de téléphone, l'image, les horaires d'ouverture et de fermeture ainsi que le statut du partenaire, puisqu'ils n'affectent évidemment pas le processus de recommandation, comme illustré dans la figure 4.25.

Entrée [2]: partners = pd.read_csv('thunder_suppliers.csv')									
Entrée [3]: partners.head(8)									
out[3]:									
		id	name	description	email	phone	image	long	lat
0	1	THE DEALER	Bar, Lounge, Drinks, Nightlife, Craft cocktail...	thedealer@gmail.com	54786954	received_262472783036295-1582349938_16866694688...	10.605023	35.840988	Rue Youssef Saheb Taba - prêt rond point Mashm...
1	2	PIZZA & GO	Pizza, Fast food, Takeaway, Quick bites, Deliv...	pizzaandgo@gmail.com	54698758	received_262472783036295-1582349938_16866694688...	10.598080	35.883133	Cité Nozha Kantaoui Hmme Sousse à côté La Vill...
2	3	MEXICO LINDO	Mexican cuisine, Authentic flavors, Tacos, Bur...	mexicolindo@gmail.com	22587457	received_262472783036295-1582349938_16866694688...	10.605892	35.867151	Rue 14 Janvier - en face Ooredoo avant rond po...
3	4	CHIKHA PIZZA & WAFFLES	Pizza, Waffles, Fast food, Desserts, Family-fr...	chikhapizzaandwaffles@gmail.com	96587144	received_262472783036295-1582349938_16866694688...	10.613410	35.852094	Avenue des orangers - à côté Café Route 66 - ...
4	6	NAPOLI PIZZA	Pizza, Italian cuisine, Pasta, Authentic flavo...	napolipizza@gmail.com	58471225	received_262472783036295-1582349938_16866694688...	10.592469	35.833727	Rue Yesser Arafat - à côté Carré Noir - Sahlou...

FIGURE 4.25 – Colonnes supprimées du jeu de données

En conséquence, nous avons conservé l'id, le nom, la description, l'adresse et la catégorie du partenaire, puisque ces champs sont pertinents et ont une influence directe sur la recommandation, comme illustré dans la figure 4.26.

Entrée [5]: partners = partners[['id','name','description','adress','category']]					
Entrée [6]: partners.head(20)					
out[6]:					
		id	name	description	adress
0	1	THE DEALER	Bar, Lounge, Drinks, Nightlife, Craft cocktail...	Rue Youssef Saheb Taba - prêt rond point Mashm...	RESTAURANT
1	2	PIZZA & GO	Pizza, Fast food, Takeaway, Quick bites, Deliv...	Cité Nozha Kantaoui Hmme Sousse à côté La Vill...	RESTAURANT
2	3	MEXICO LINDO	Mexican cuisine, Authentic flavors, Tacos, Bur...	Rue 14 Janvier - en face Ooredoo avant rond po...	RESTAURANT
3	4	CHIKHA PIZZA & WAFFLES	Pizza, Waffles, Fast food, Desserts, Family-fr...	Avenue des orangers - à côté Café Route 66 - ...	RESTAURANT
4	6	NAPOLI PIZZA	Pizza, Italian cuisine, Pasta, Authentic flavo...	Rue Yesser Arafat - à côté Carré Noir - Sahlou...	RESTAURANT
5	7	BONCONCINI	Italian cuisine, Cheese, Fine dining, Pasta, W...	Rue Yasser Arafat - Immeuble Hasdrubal - Sahlou...	RESTAURANT
6	8	BRUSCHETTA	Italian cuisine, Bruschetta, Appetizers, Wine ...	Rue Yasser Arafat - en face Zaitouna bank - Sa...	RESTAURANT
7	10	ZEUS BY ARGOS	Greek cuisine, Mediterranean flavors, Gyros, S...	Rue Imem Moslem - en face Dar El Kunafa - Khz...	RESTAURANT
8	11	TRATTORIA DA FRANCO	Italian cuisine, Pasta, Traditional dishes, Fa...	Rue Sadek Mkaadem - prêt de Café Don Pablo - Sa...	RESTAURANT
9	12	PIZZERIA BIANCO	Pizza, Italian cuisine, Wood-fired oven, Neapo...	Taher sfar	RESTAURANT
10	13	FRESH JUICE & CRÊPE	Juice bar, Smoothies, Crepes, Healthy options,...	Rue Sadek Laroussi - prêt Café My Time - Sousse	PASTRIES
11	14	Papillon pizza by chef sousse	Pizza, Gourmet, Chef's specials, Upscale dinin...	avenue des orangers	RESTAURANT
12	15	kfc	Fast food, Fried chicken, Burgers, Drive-thru...	Taher sfar	RESTAURANT
13	21	TANDO'S LA CARTE	Indian cuisine, Tandoori dishes, Curries, À la...	Rue Hassan Ibn Said - prêt Café Montecelio - K...	RESTAURANT
14	22	MR FOOD	Fast food, Burgers, Sandwiches, Quick bites, V...	Rue Charles De Gaulle - prêt Dar Chabeb Hmme S...	RESTAURANT
15	23	LE MEXICAIN TACOS ET BURRITOS	Mexican cuisine, Tacos, Burritos, Spicy flavor...	Rue Jalloul Ben Cherifa - prêt Café Black Box ...	RESTAURANT
16	24	K-FOOD	Korean cuisine, Bibimbap, Kimchi, BBQ, Banchan...	Rue Sadek Mkaadem - à côté Café White House - S...	RESTAURANT
17	25	CHEDEN FOOD	Local cuisine, Traditional dishes, Authentic f...	Rue Imem Moslem - prêt Dar El Kunafa - Khzema ...	RESTAURANT
18	26	BANKSY PIZZA	Pizza, Artistic theme, Street food, Creative t...	Rue De Matmata - à côté Café Montecelio - Khze...	RESTAURANT
19	27	TAQIMA FOOD	Tacos, pizza, Tacos à la carte, Quesadillas, ...	Djebel Mzaouia - à côté de la station-service - LOCAL PRODUCE	RESTAURANT

FIGURE 4.26 – Colonnes conservées du jeu de données

- **Aggrégation par colonne** : dans le contexte de prétraitement de nos données, nous avons utilisé un ensemble de fonctions pour regrouper toutes les données pertinentes dans la même colonne que nous avons appelée "tags". Nous avons adopté cette approche car nous souhaitions représenter toutes les données caractérisant le partenaire dans une seule colonne.

```

2 3 MEXICO LINDO Mexican cuisine, Authentic flavors, Tacos, Bur... [Rue 14 Janvier , en face Ooredoo avant rond p... [RESTAURANT]
3 4 CHIKHA PIZZA & WAFFLES Pizza, Waffles, Fast food, Desserts, Family-fr... [ Avenue des oranges , à côté Café Route 66 ... [RESTAURANT]
4 6 NAPOLI PIZZA Pizza, Italian cuisine, Pasta, Authentic flavo... [Rue Yesser Arafat , à côté Carré Noir , Sahlo... [RESTAURANT]

Entrée [14]: partners['description'] = partners['description'].apply(convert)
partners.head()

Out[14]:
   id          name      description           adress      category
0  1  THE DEALER  [Bar, Lounge, Drinks, Nightlife, Craft cocktail...  [Rue Youssef Saheb Taba , près rond point Mash... [RESTAURANT]
1  2    PIZZA & GO  [Pizza, Fast food, Takeaway, Quick bites, Deli...  [Cité Nozha Kantaoui Hmme Sousse à côté La Vil... [RESTAURANT]
2  3  MEXICO LINDO  [Mexican cuisine, Authentic flavors, Tacos, Bur...  [Rue 14 Janvier , en face Ooredoo avant rond p... [RESTAURANT]
3  4  CHIKHA PIZZA & WAFFLES  [Pizza, Waffles, Fast food, Desserts, Family-fr...  [ Avenue des oranges , à côté Café Route 66 ... [RESTAURANT]
4  6  NAPOLI PIZZA  [Pizza, Italian cuisine, Pasta, Authentic flavo...  [Rue Yesser Arafat , à côté Carré Noir , Sahlo... [RESTAURANT]

Entrée [15]: partners['tags'] = partners['description'] + partners['adress'] + partners['category']

Entrée [16]: new = partners.drop(columns=['description', 'adress', 'category'])
#new.head()

Entrée [17]: new['tags'] = new['tags'].str.replace('[', ' ', regex=False).str.replace(']', ' ', regex=False)
new.head()

Out[17]:
   id          name      tags
0  1  THE DEALER  Bar, Lounge, Drinks, Nightlife, Craft cocktail...
1  2    PIZZA & GO  Pizza, Fast food, Takeaway, Quick bites, Deliv...
2  3  MEXICO LINDO  Mexican cuisine, Authentic flavors, Tacos, Bur...
3  4  CHIKHA PIZZA & WAFFLES  Pizza, Waffles, Fast food, Desserts, Family-fr...
4  6  NAPOLI PIZZA  Pizza, Italian cuisine, Pasta, Authentic flavo...

```

FIGURE 4.27 – Aggrégation par colonne

- **Conversion en vecteurs** : une fois que nous avons agrégé les données pertinentes par colonne, nous pouvons convertir ces colonnes en vecteurs représentant les profils des partenaires.

```

Entrée [19]: from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features=100, stop_words='english')

Entrée [20]: vector = cv.fit_transform(new['tags']).toarray()

Entrée [21]: vector.shape

Out[21]: (114, 100)

```

FIGURE 4.28 – Conversion en vecteurs

- **Choix de l'algorithme :**

La **similarité cosinus**, mesure la similarité entre deux vecteurs dans un espace vectoriel. Elle est souvent utilisée dans le traitement du langage naturel et l'apprentissage automatique. La similarité cosinus évalue l'alignement directionnel des vecteurs plutôt que leur magnitude. Elle est calculée à l'aide du produit scalaire et des normes euclidiennes des vecteurs. La similarité cosinus varie entre -1 et 1, où 1 indique une similarité parfaite, -1 une similarité négative parfaite et 0 une similarité nulle. Cette mesure est utilisée pour comparer des documents et des mots dans des tâches telles que la recherche d'informations, la recommandation de produits et la classification de texte.

Avec nos vecteurs de données représentant les partenaires, nous calculons la similarité cosinus entre eux afin de trouver des profils similaires et de recommander des partenaires en fonction de la similarité de leurs vecteurs.

```

Entrée [22]: from sklearn.metrics.pairwise import cosine_similarity
Entrée [23]: similarity = cosine_similarity(vector)
Entrée [24]: print(similarity)

[[1.          0.12309149 0.32732684 ... 0.          0.          0.          ]
 [0.12309149 1.          0.24174689 ... 0.          0.          0.          ]
 [0.32732684 0.24174689 1.          ... 0.          0.          0.          ]
 ...
 [0.          0.          0.          ... 1.          0.4472136  0.35355339]
 [0.          0.          0.          ... 0.4472136  1.          0.31622777]
 [0.          0.          0.          ... 0.35355339  0.31622777 1.          ]]
]

Entrée [25]: def recommend(partner):
    index = new[new['name'] == partner].index[0]
    distances = sorted(list(enumerate(similarity[index])), reverse=True, key=lambda x: x[1])
    for i in distances[1:11]:
        partner_name = new.iloc[i[0]]['name']
        print(partner_name)

Entrée [26]: recommend('PAPADAM KHZEMA')

PAPADAM FOOD
EZZAYET
PAPADAM SAHLOUL
BORGÏ
ABDALLAH CHANEK SAHLOUL
Cool Tounsi
BAZGÛT
CHEDEN FOOD
STOUFA FOOD
K-FOOD

```

FIGURE 4.29 – L'algorithme similarité cosinus

Nous avons également utilisé l'algorithme des K plus proches voisins (KNN) pour tester nos données. Malheureusement, les résultats obtenus n'étaient pas aussi satisfaisants que ceux obtenus avec l'algorithme de similarité cosinus. En effet, l'algorithme de similarité cosinus a démontré des performances supérieures et s'est avéré mieux adapté à nos besoins spécifiques.

- **Évaluez les performances** : nous avons utilisé la mesure de précision pour tester notre système.

```

Entrée [32]: cosine_similarity_score = similarity[0, 0]
# Calculate precision
precision = (cosine_similarity_score + 1) / 2
# Print the precision
print("Cosine similarity precision:", precision)
Cosine similarity precision: 0.8

```

FIGURE 4.30 – Mesure de précision

- **Intégration de système dans notre application** : Nous avons utilisé Flask pour créer une API permettant l'intégration de notre système de recommandation. Nous avons établi une communication entre cette API Flask et l'API Laravel de notre application. Maintenant, lorsque l'application mobile s'ouvre, elle envoie une requête à son API Laravel, qui à son tour envoie une requête à l'API Flask. La réponse obtenue suit le même schéma. La figure 4.31, explique ce processus.

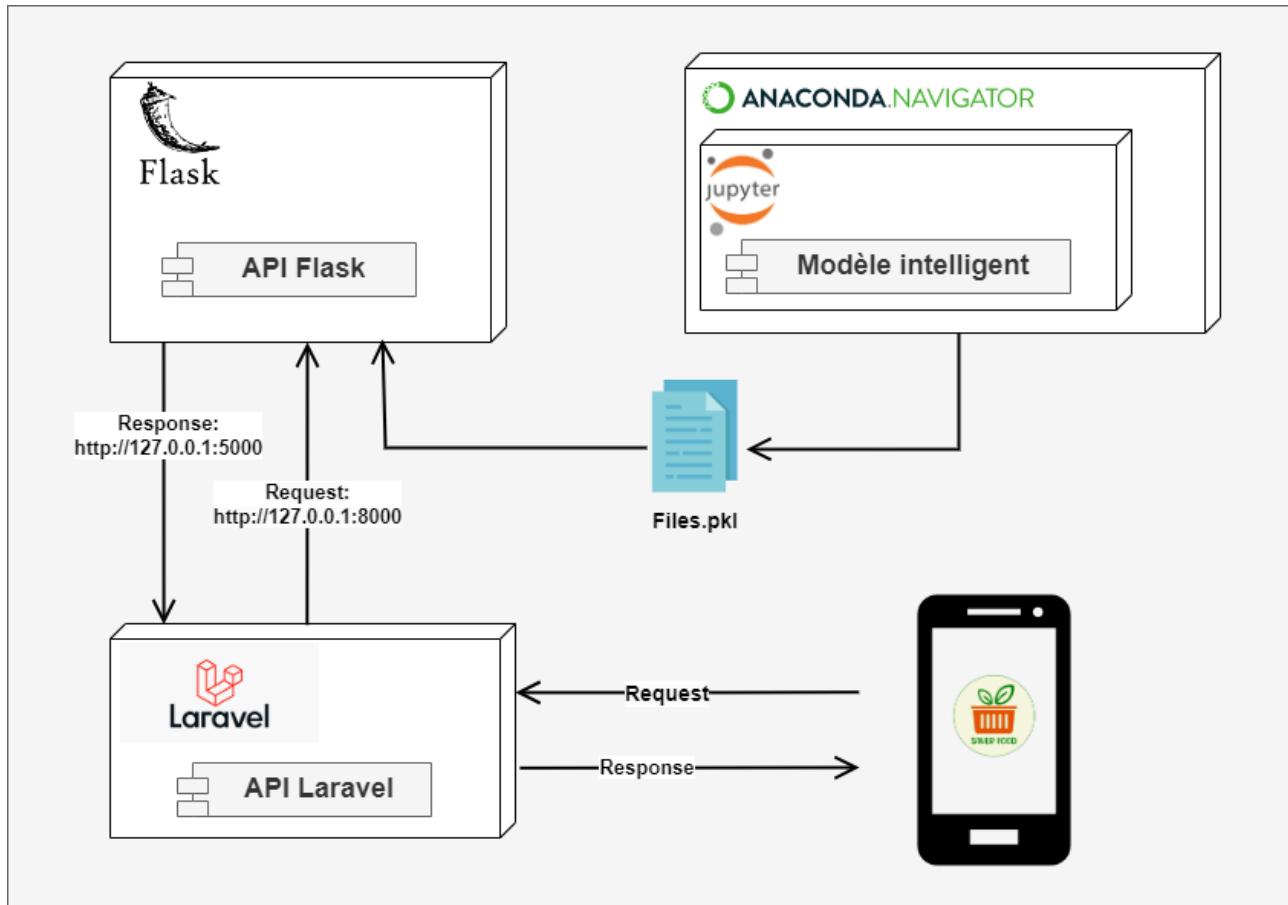


FIGURE 4.31 – Processus d'intégration de système de recommandation

Grâce à ce système de recommandation intelligent, nous espérons offrir une expérience d'achat plus agréable et pertinente à nos clients, tout en augmentant nos ventes en proposant des boxes qui correspondent véritablement à leurs attentes.

2. Construction du système créé avec Neo4j :

Neo4j est une base de données de graphes hautement performante et flexible. Elle permet de stocker, de gérer et d'interroger des données sous forme de graphes, où les entités sont représentées par des nœuds et les relations entre elles par des arêtes. Neo4j offre des fonctionnalités avancées pour la modélisation, la navigation et l'analyse des relations complexes entre les données. Grâce à son langage de requête spécifique aux graphes, appelé Cypher, il est possible d'effectuer des recherches et des calculs avancés sur les données connectées. Neo4j est utilisé dans divers domaines tels que les réseaux sociaux, la recommandation de produits et la gestion des connaissances.

Puisque nous avons réalisé le système de recommandation basé sur les interactions de l'utilisateur, nous avons décidé de développer un autre système avec Neo4j pour les nouveaux utilisateurs. Étant donné qu'ils n'ont pas d'historique sur l'application, ce système recommande des "boxes" adaptées à ces nouveaux clients en se basant sur l'historique des autres utilisateurs de l'application. Plus précisément, notre système recommande à un nouveau client les catégories de boxes préférées par les clients du même âge et du même sexe.

Pour mettre en place ce système, nous avons suivi les étapes suivantes de manière séquentielle comme indiqué dans la figure 4.32 :

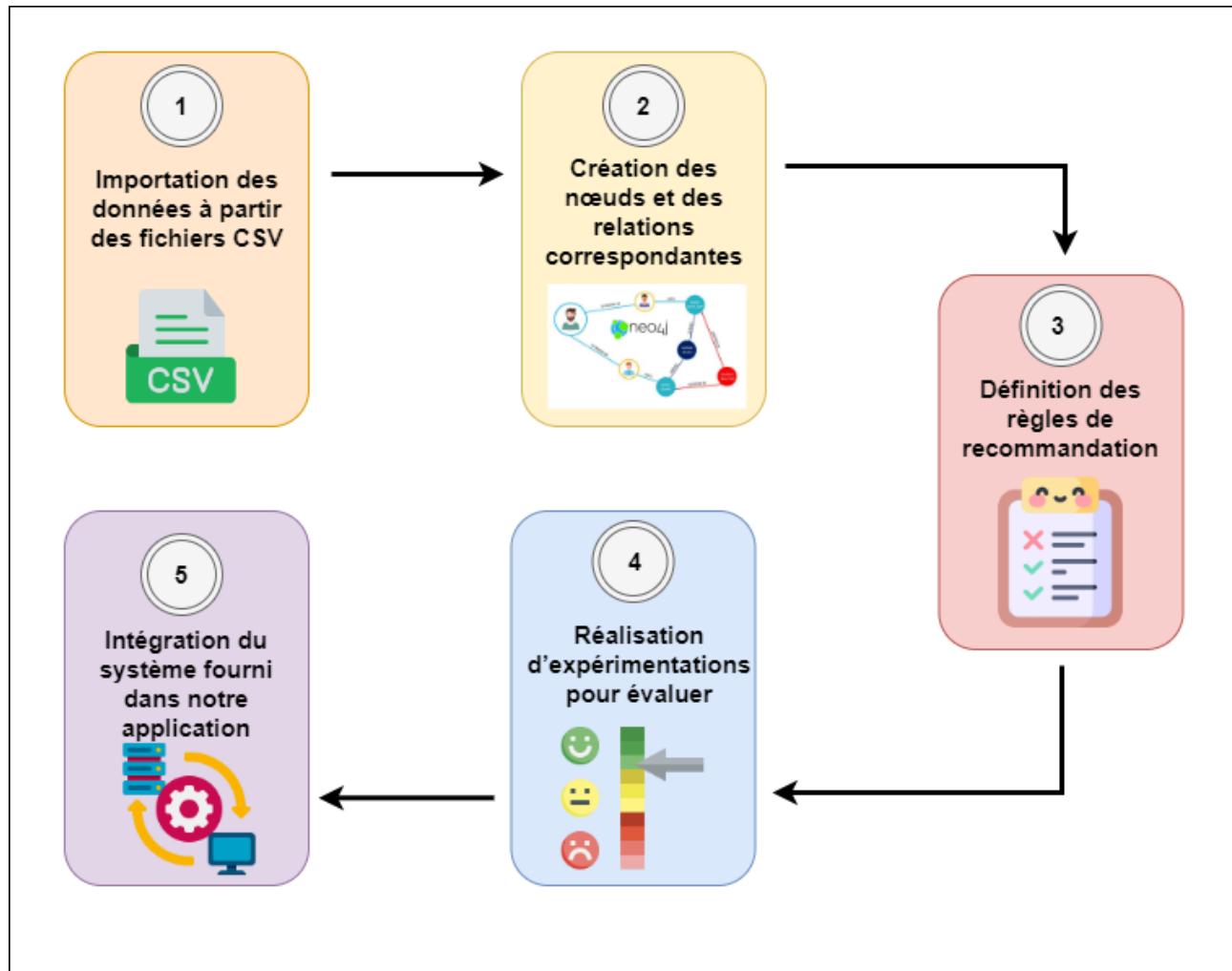


FIGURE 4.32 – Graphe du système créé avec Neo4j

La figure 4.33 illustre la représentation graphique de notre graphe.



FIGURE 4.33 – Graphe du système créé avec Neo4j

Conclusion

Ce dernier chapitre porte sur la réalisation de notre travail, il nous a permis de présenter l'environnement matériel et logiciel de notre projet, il nous a aussi permis de présenter les différentes technologies que nous avons utilisées pour l'aboutissement de ce travail. Et enfin, nous avons publié le travail réalisé au cours de chaque sprint.

Conclusion générale et perspectives

Le présent rapport a été rédigé dans le cadre de mon projet de fin d'études pour l'obtention du diplôme d'ingénieur en informatique, spécialité génie logiciel, réalisé au sein de la société Webify.

L'objectif de ce stage était de développer une plateforme visant à lutter contre le gaspillage alimentaire. Cette plateforme vise à générer des bénéfices environnementaux, économiques et sociaux en contribuant à préserver les ressources et à sensibiliser le grand public. Il s'agit d'un outil innovant qui favorise un avenir plus durable en encourageant une consommation responsable et en mobilisant la communauté pour agir contre le gaspillage alimentaire.

commencé/débuté

Pour atteindre notre objectif, nous avons entamé par une étude préliminaire approfondie afin de bien comprendre le contexte du projet. Nous avons également identifié les besoins fonctionnels et non fonctionnels, ce qui nous a permis de réaliser une étude conceptuelle en modélisant notre système à l'aide de diagrammes de cas d'utilisation et de séquences systèmes.

Ensuite, nous avons élaboré la conception détaillée de notre projet en commençant par une architecture globale présentant les différents patrons que nous avons utilisés. Nous avons ensuite développé des diagrammes de classes d'entités pour représenter les différentes entités et relations du système. Par la suite, nous avons décrit l'architecture dynamique de l'application à travers des diagrammes de séquences, de classes participantes, de classes de conception, d'activité et d'état-transition.

Enfin, nous avons abordé l'étape de la réalisation où nous avons traduit la conception théorique en une implémentation physique. Cette phase a été l'occasion d'appliquer nos compétences techniques et de mettre en pratique les concepts que nous avons étudiés.

Ce stage a constitué une expérience extrêmement enrichissante, tant sur le plan technique que professionnel. Sur le plan technique, il nous a permis d'explorer de nouveaux domaines de l'informatique en plein essor, tels que l'apprentissage automatique et de maîtriser des technologies avancées telles que Neo4j.

Sur le plan professionnel, ce projet a été une véritable opportunité de travailler au sein d'une entreprise et de vivre une expérience concrète de développement logiciel. Il nous a également permis de renforcer nos compétences en gestion de projet, en communication et en collaboration au sein d'une équipe multidisciplinaire.

En conclusion, les fonctionnalités de notre logiciel répondent parfaitement aux besoins de

notre client. Néanmoins, nous pouvons prochainement :

- Concevoir et mettre en place un système de notifications avancé qui permettra de communiquer efficacement avec les utilisateurs en temps réel.
- Nous souhaitons améliorer et optimiser notre système de recommandations afin de garantir une précision maximale et de nous adapter en permanence aux préférences changeantes de nos clients. De plus, nous avons l'intention de constituer un ensemble de données spécifique à notre application pour assurer la qualité et la pertinence des recommandations que nous offrons.

Références

- [1] *Android Studio site web : Blogdumoderateur [accès le 22/05/2023]*. URL : www.blogdumoderateur.com/tools/visual-studio-code/.
- [2] *Angular site web : Bocasay [accès le 22/05/2023]*. URL : www.bocasay.com/fr/developpement-application-angular.
- [3] *Dart site web : Dart [accès le 22/05/2023]*. URL : dart.dev/overview.
- [4] *Diagramme de séquence site web : Lucidchart [accès le 4/06/2023]*. URL : www.lucidchart.com/pages/fr/diagramme-de-sequence-uml.
- [5] *Flask site web : Wikipedia [accès le 22/06/2023]*. URL : [fr.wikipedia.org/wiki/Flask_\(framework\)](http://fr.wikipedia.org/wiki/Flask_(framework)).
- [6] *Flutter site web : Mobizel [accès le 22/05/2023]*. URL : www.mobizel.com/flutter-avantages-inconvenients/.
- [7] *Git site web : Edutechwiki [accès le 22/05/2023]*. URL : edutechwiki.unige.ch/fr/Git_et_Github.
- [8] *Json site web : MDN Plus [accès le 22/05/2023]*. URL : developer.mozilla.org/fr/docs/Learn/JavaScript/Objects/JSON.
- [9] *Langage de modélisation UML site web : Digital Guide IONOS [accès le 30/05/2023]*. URL : <https://www.ionos.fr/digitalguide/sites-internet/developpement-web/uml-un-langage-de-modelisation-pour-la-programmation-orientee-objet/>.
- [10] *Laravel site web : Kinsta [accès le 22/05/2023]*. URL : kinsta.com/fr/base-de-connaissances/qu-est-ce-que-laravel/.
- [11] *Middleware site web : ApprendreLaravel [accès le 06/06/2023]*. URL : www.apprendre-laravel.fr/laraguide/2017-11-30-proteger-nos-pages-avec-un-middleware.
- [12] *MySQL site web : Wikipedia [accès le 22/05/2023]*. URL : fr.wikipedia.org/wiki/MySQL.
- [13] *Neo4j site web : d-booker [accès le 22/06/2023]*. URL : www.d-booker.fr/content/68-introduction-bases-de-donnees-neo4j.
- [14] *PHP site web : Dart [accès le 22/05/2023]*. URL : fr.wikipedia.org/wiki/PHP.
- [15] *POSTMAN site web : Welovedevs [accès le 22/05/2023]*. URL : <https://welovedevs.com/fr/articles/postman/>.
- [16] *Python site web : Wikipedia [accès le 22/06/2023]*. URL : [https://fr.wikipedia.org/wiki/Python_\(langage\)](https://fr.wikipedia.org/wiki/Python_(langage)).
- [17] *TypeScript site web : Wikipedia [accès le 22/05/2023]*. URL : fr.wikipedia.org/wiki/TypeScript.

- [18] *Visual Studio Code site web : Wikipedia [accès le 22/05/2023]. URL : www.blogdumoderateur.com/tools/visual-studio-code/.*