

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
**Université A. Mira de Béjaïa**  
Faculté des Sciences Exactes  
Département d'Informatique



## Mémoire de Fin de Cycle

En vue de l'obtention du diplôme de Master en Informatique  
Option : Génie Logiciel

Thème

---

**Conception et réalisation d'une application  
web pour la Gestion de la relation client d'une Agence de Voyage**  
*Cas Ziguade Tour*

---

**Réalisé par :**  
Mlle ZENATI Lina  
Mlle TIAB Mélissa Smicha

**Soutenu le : 1 juillet 2025**

**Devant le jury composé de :**

Président : M. ZERARGA Lotfi Maître de Conférences A Université A. Mira Béjaïa  
Examinatrice : Mme GADOUCHE Hania Maître de Conférences B Université A. Mira Béjaïa  
Examinatrice : Mme GHANEM Souhila Maître de Conférences A Université A. Mira Béjaïa  
Examinatrice : Mme AZIZOU Zahia Maître de Conférences B Université A. Mira Béjaïa  
Encadrant : M. SIDER Abderrahmane Maître de Conférences A Université A. Mira Béjaïa

**Promotion 2024–2025**

## Dédicaces Lina

*Je dédie ce modeste travail :*

À mes chers parents, pour leur amour inconditionnel, leurs sacrifices silencieux et leur soutien indéfectible, source constante de motivation et de courage.

À mes frères Farid, Nadir et Faouzi, ainsi qu'à leurs épouses, pour leur présence, leur affection et leurs encouragements dans toutes les étapes de ma vie.

À ma sœur Sadjia et son mari, pour leur tendresse, leur bienveillance et leur réconfort.

À ma chère Aïcha, qui m'a tant aidée et soutenue avec générosité, patience et bienveillance.

À mes tantes, pour leurs prières sincères et leurs encouragements chaleureux.

À toute ma famille, pilier précieux de ma réussite.

À tous mes amis et proches, fidèles et bienveillants, qui m'ont encouragée tout au long de ce parcours.

À mon encadrant, Monsieur **Sider Abderrahmane**, pour sa disponibilité, ses conseils avisés et son accompagnement tout au long de ce travail.

**Lina**

## Dédicaces Mélissa

*Je dédie ce modeste travail :*

À mes parents, pour leur amour inconditionnel, leur soutien sans faille et leurs encouragements constants. Vous avez été ma source d'inspiration et de motivation tout au long de ce parcours.

À ma sœur Nihad, pour son amour, son soutien constant et ses encouragements. Ta bienveillance m'a portée tout au long de ce projet.

À mon frère Abdou, pour sa présence rassurante et ses mots motivants. Merci pour ton soutien tout au long de cette aventure.

À mes professeurs, pour leur dévouement et leur enseignement précieux tout au long de mon cursus.

À Monsieur Sider Abderrahmane, mon encadrant, pour ses conseils précieux, sa disponibilité et son accompagnement tout au long de ce projet. Votre encadrement a été une véritable source d'enrichissement.

À tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail, je vous exprime ma plus profonde gratitude.

**Mélissa**

# Remerciements

Nous tenons tout d'abord à exprimer notre profonde gratitude à Dieu Tout-Puissant, qui nous a donné la force, la patience et la persévérance nécessaires pour mener à bien ce travail.

Nous adressons nos remerciements les plus sincères à notre encadrant, Monsieur Sider Abderrahmane, pour sa disponibilité, ses conseils pertinents, son suivi rigoureux ainsi que pour sa confiance et son accompagnement tout au long de ce projet.

Nous remercions également l'ensemble des enseignants du Département d'Informatique de l'Université de Béjaïa pour la qualité de l'enseignement dispensé tout au long de notre parcours universitaire.

Nous exprimons aussi notre profonde reconnaissance aux membres du jury, pour avoir accepté d'examiner et d'évaluer notre travail avec bienveillance et attention.

Nos remerciements vont également à toutes les personnes qui nous ont aidées de près ou de loin dans la réalisation de ce mémoire.

Nous exprimons toute notre reconnaissance à nos familles, pour leur soutien moral et matériel, leur patience et leur amour inconditionnel.

Enfin, nous remercions nos amis et camarades pour leur aide, leur bienveillance et les bons moments partagés durant ce cursus.

**Zenati Lina    &    Tiab Mélissa Smicha**

# Table De matières

<b>Introduction Générale</b>	<b>1</b>
<b>Chapitre 1 : Cadre du travail et Problématique</b>	<b>3</b>
1.1 Introduction . . . . .	3
1.2 Définition et concepts du tourisme . . . . .	3
1.2.1 Définition du tourisme . . . . .	3
1.2.2 Typologie du tourisme . . . . .	3
1.3 Présentation de l'organisme d'accueil . . . . .	4
1.3.1 Présentation de ZIGUADE TOUR . . . . .	4
1.3.2 Historique de l'agence ZIGUADE TOUR . . . . .	5
1.4 L'étude de l'existant . . . . .	5
1.4.1 Description des applications existantes . . . . .	5
1.4.2 La comparaison des services et technologies des applications web . . . . .	8
1.4.3 L'analyse des performances . . . . .	10
1.4.4 Modes de Réservation et de Paiement . . . . .	13
1.4.5 Services et offres . . . . .	14
1.4.6 Critique de l'existant . . . . .	16
1.5 Problématique et objectifs du travail . . . . .	16
1.5.1 Problématique . . . . .	16
1.5.2 Solution proposée . . . . .	17
1.6 Cahier des charges . . . . .	18
1.6.1 Besoins fonctionnels . . . . .	18
1.6.2 Besoins non fonctionnels . . . . .	22
1.7 Conclusion . . . . .	23
<b>Chapitre 2 : Spécification des besoin</b>	<b>24</b>
2.1 Introduction . . . . .	24
2.2 Méthodologie et Conception . . . . .	24
2.3 Identification des acteurs . . . . .	24
2.4 Gestion de projet avec scrum . . . . .	25
2.4.1 Les rôles . . . . .	25
2.4.2 Définition des User story . . . . .	25
2.4.3 Backlog product . . . . .	30
2.4.4 Sprint Backlog . . . . .	31
2.5 Les Outils et Langages utilisés . . . . .	36

2.6	Conclusion . . . . .	38
<b>Chapitre 3 : Etude Sprint 1,2 et 3</b>		<b>39</b>
3.1	Introduction . . . . .	39
3.2	Étude du 1er Sprint . . . . .	39
3.2.1	Décomposition du Sprint 1 . . . . .	40
3.2.2	Diagramme de cas d'utilisation global du Sprint 1 . . . . .	40
3.2.3	Cas d'utilisation Inscription . . . . .	42
3.2.4	Cas d'utilisation Authentification . . . . .	43
3.2.5	Cas d'utilisation Gérer mon compte . . . . .	45
3.2.6	Cas d'utilisation Ajouter un agent . . . . .	47
3.2.7	Cas d'utilisation Modifier un agent . . . . .	49
3.2.8	Cas d'utilisation Supprimer un utilisateur inscrit . . . . .	51
3.2.9	Diagramme de Classe premier sprint . . . . .	53
3.2.10	Interfaces du premier sprint . . . . .	54
3.3	Etude de 2ème Sprint . . . . .	57
3.3.1	Diagramme de cas d'utilisation Globale Sprint 2 . . . . .	59
3.3.2	Cas d'utilisation Créer un voyage . . . . .	61
3.3.3	Cas d'utilisation Modifier un voyage . . . . .	62
3.3.4	Cas d'utilisation Publier un Voyage sur Facebook . . . . .	64
3.3.5	Cas d'utilisation Récupérer les nouveaux Commentaires . . . . .	66
3.3.6	Cas d'utilisation Sélectionner des Commentaires . . . . .	68
3.3.7	Cas d'utilisation Réserver une Omra . . . . .	70
3.3.8	Cas d'utilisation Filtrer une Réservation . . . . .	72
3.3.9	Cas d'utilisation Rechercher un vol . . . . .	74
3.3.10	Cas d'utilisation Rechercher un Hôtel . . . . .	76
3.3.11	Cas d'utilisation Payer avec Chargily . . . . .	78
3.3.12	Cas d'utilisation Envoyer les factures par email . . . . .	80
3.3.13	Cas d'utilisation Configurer les APIs . . . . .	82
3.3.14	Modèle relationnel du 2 <sup>ème</sup> sprint . . . . .	86
3.3.15	Interfaces du deuxième sprint . . . . .	86
3.4	Étude de 3ème Sprint . . . . .	95
3.4.1	Décomposition de Sprint 3 . . . . .	96
3.4.2	Diagramme de cas d'utilisation Globale Sprint 3 . . . . .	96
3.4.3	Cas d'utilisation Souscrire une assurance . . . . .	97
3.4.4	Diagramme de séquence de Cas d'utilisation souscrire une assurance . . . . .	98
3.4.5	Cas d'utilisation Consulter la liste des contrats d'assurance . . . . .	99
3.4.6	Cas d'utilisation Soumettre une demande de Visa . . . . .	101
3.4.7	Cas d'utilisation Consulter la liste des visas . . . . .	103
3.4.8	Diagramme de Classe 3ème sprint . . . . .	105
3.4.9	Modèle relationnel de 3ème sprint . . . . .	106
3.4.10	Interfaces du troisième sprint . . . . .	107
3.5	Conclusion . . . . .	110
<b>Chapitre 4 : Intégration des APIs externes : Réseaux sociaux et Paiement</b>		<b>111</b>

4.1	Introduction . . . . .	111
4.2	Intégration des API de Facebook et Instagram . . . . .	111
4.2.1	Présentation de Meta Graph API . . . . .	111
4.2.2	Conditions d'utilisation des comptes pour accéder aux APIs Meta . . . . .	112
4.2.3	Authentification et génération des jetons . . . . .	112
4.2.4	Publication et gestion des commentaires . . . . .	113
4.2.5	Limitations d'utilisation des APIs Meta (Quotas) . . . . .	114
4.2.6	Gestion des erreurs courantes avec l'API Graph . . . . .	115
4.2.7	Utilisation de Ngrok pour les tests en local . . . . .	115
4.3	Intégration des API de paiement Stripe et Chargily . . . . .	116
4.3.1	Présentation des solutions . . . . .	116
4.3.2	Inscription et génération des clés API . . . . .	116
4.3.3	Intégration technique . . . . .	117
4.3.4	Tests . . . . .	118
4.3.5	Différences principales . . . . .	118
4.3.6	Gestion des Webhooks . . . . .	119
4.3.7	Webhooks Stripe . . . . .	119
4.3.8	Webhooks Chargily . . . . .	120
4.4	Conclusion . . . . .	120
	<b>Conclusion et Perspectives</b>	<b>121</b>
	<b>Annexe</b>	<b>122</b>
	<b>Bibliographie</b>	<b>124</b>

# Liste des figures

1.1	Logo de l'organisme d'accueil ZIGUADE TOUR . . . . .	4
1.2	Site de Gouraya Tour . . . . .	5
1.3	Site de Meryam Tour . . . . .	6
1.4	Page d'accueil du site web Dune Voyages . . . . .	6
1.5	Page d'accueil du site web H24Voyages . . . . .	7
1.6	Page d'accueil du site web Fram . . . . .	7
1.7	Page d'accueil du site web Booking . . . . .	8
1.8	Logo de l'outil Wappalyzer . . . . .	9
1.9	page d'accueil du site web site-analyzer . . . . .	11
3.1	diagramme de cas d'utilisation sprint1 . . . . .	41
3.2	Diagramme de Séquence Inscription . . . . .	43
3.3	Diagramme de Séquence Authentification . . . . .	45
3.4	Diagramme de Séquence Gérer mon compte . . . . .	47
3.5	Diagramme de Séquence Ajouter un agent . . . . .	49
3.6	Diagramme de Séquence Modifier un agent . . . . .	51
3.7	Diagramme de Séquence Supprimer un utilisateur inscrit . . . . .	53
3.8	Diagramme de Classe Sprint 1 . . . . .	54
3.9	Interface De la page d'accueil . . . . .	55
3.10	Interface d'inscription utilisateur . . . . .	55
3.11	Interface d'authentification utilisateur . . . . .	56
3.12	Interface de gestion de l'espace utilisateur . . . . .	56
3.13	Interface d'ajout d'un agent . . . . .	57
3.14	Interface des utilisateurs inscrits . . . . .	57
3.15	Diagramme de cas d'utilisation sprint2 . . . . .	60
3.16	Diagramme de Séquence Créer un Voyage . . . . .	62
3.17	Diagramme de Séquence Modifier un voyage . . . . .	64
3.18	Diagramme de Séquence Publier un Voyage sur facebook . . . . .	66
3.19	Diagramme de séquence - Récupérer les nouveaux commentaires . . . . .	68
3.20	Diagramme de séquence - Sélectionner des commentaires . . . . .	70
3.21	Diagramme de séquence - Réserver une Omra . . . . .	71
3.22	Diagramme de séquence - Filtrer une réservation . . . . .	74
3.23	Diagramme de séquence - Rechercher un vol . . . . .	76
3.24	Diagramme de séquence - Rechercher un hôtel . . . . .	78
3.25	Diagramme de séquence - Payer avec Chargily . . . . .	80
3.26	Diagramme de séquence - Envoyer factures par email . . . . .	82

3.27	Diagramme de séquence - Configurer les APIs . . . . .	84
3.28	Diagramme de classe - Deuxième sprint . . . . .	85
3.29	Interface de création d'un voyage . . . . .	86
3.30	Interface de publication d'un voyage . . . . .	87
3.31	Interface liste de publication pour chaque offre . . . . .	87
3.32	Interface de récupération des commentaires . . . . .	88
3.33	Interface de publication de voyage sur le site . . . . .	88
3.34	Interface de publication de voyage sur le site . . . . .	89
3.35	Interface de capture de la publication sur facebook . . . . .	89
3.36	Interface de capture de la publication sur instagram . . . . .	90
3.37	Interface liste de voyage sur site avec commentaire sélectionner et nombre de likes . . . . .	90
3.38	Interface liste de voyage sur site avec commentaire sélectionner et nombre de likes . . . . .	91
3.39	Interface Filtrer les réservations . . . . .	91
3.40	Interface de recherche de vols . . . . .	92
3.41	Interface de recherche d'hôtels . . . . .	92
3.42	interface choix de paiement . . . . .	93
3.43	Interface de paiement via Chargily . . . . .	93
3.44	Interface de paiement avec Stripe . . . . .	94
3.45	Interface Mes factures . . . . .	95
3.46	Interface de configuration des APIs . . . . .	95
3.47	Répartition temporelle du sprint 3 . . . . .	96
3.48	Diagramme de cas d'utilisation - 3ème sprint . . . . .	97
3.49	Diagramme de séquence - Souscrire une assurance . . . . .	99
3.50	Diagramme de séquence - Consulter la liste des contrats d'assurance . . . . .	101
3.51	Diagramme de séquence - Soumettre une demande de Visa . . . . .	103
3.52	Diagramme de séquence - Consulter la liste des visas . . . . .	105
3.53	Diagramme de classe - 3ème sprint . . . . .	106
3.54	Interface de souscription à une assurance . . . . .	107
3.55	Interface de consultation des assurances . . . . .	108
3.56	Interface de soumission d'une demande de visa . . . . .	108
3.57	Interface de consultation des offres de visa . . . . .	109
3.58	Interface de formulaire de contact . . . . .	109
3.59	Interface de répondre a un message . . . . .	110
4.1	Diagramme de cas d'utilisation global de l'application . . . . .	122
4.2	Diagramme de classe global de l'application . . . . .	123

# Liste des tableaux

1.1	Comparaison des technologies des applications web . . . . .	10
1.2	comparaison des applications web . . . . .	12
1.3	comparaison des modes de réservation et de filtrage des applications web . .	14
1.4	Comparaison des services des applications web . . . . .	15
1.5	Avantages et inconvénients des sites web de réservation . . . . .	16
2.1	Rôles et actions des différents acteurs de l'application . . . . .	25
2.2	Répartition des rôles Scrum pour le projet . . . . .	25
2.3	Product Backlog de l'application ZIGUADE TOUR . . . . .	31
2.4	Sprint Backlog . . . . .	36
3.1	Description textuelle du cas d'utilisation «Inscription » . . . . .	42
3.2	Description textuelle du cas d'utilisation « Authentification » . . . . .	44
3.3	Description textuelle du cas d'utilisation « Gérer mon compte » . . . . .	46
3.4	Description textuelle du cas d'utilisation « Ajouter un agent » . . . . .	48
3.5	Description textuelle du cas d'utilisation « Modifier un agent » . . . . .	50
3.6	Description textuelle du cas d'utilisation « Supprimer un utilisateur inscrit »	53
3.7	Description textuelle du cas d'utilisation « Créer un voyage » . . . . .	61
3.8	Description textuelle du cas d'utilisation « Modifier un voyage » . . . . .	63
3.9	Description textuelle du cas d'utilisation « Publier un Voyage sur Facebook »	65
3.10	Description textuelle du cas d'utilisation « Récupérer les nouveaux Commén- taires » . . . . .	67
3.11	Description textuelle du cas d'utilisation Sélectionner des Commentaires . .	69
3.12	Description textuelle du cas d'utilisation Réserver une Omra . . . . .	71
3.13	Description textuelle du cas d'utilisation « Filtrer les réservations » . . . . .	73
3.14	Description textuelle du cas d'utilisation « Rechercher un vol » . . . . .	75
3.15	Description textuelle du cas d'utilisation « Rechercher un hôtel » . . . . .	77
3.16	Description textuelle du cas d'utilisation « Payer avec Chargily » . . . . .	79
3.17	Description textuelle du cas d'utilisation « Envoyer une facture par email » .	81
3.18	Description textuelle du cas d'utilisation « Configurer les APIs » . . . . .	83
3.19	Description textuelle du cas d'utilisation Souscrire une assurance . . . . .	98
3.20	Description textuelle du cas d'utilisation Consulter la liste des contrats d'as- surance . . . . .	100
3.21	Description textuelle du cas d'utilisation Soumettre une demande de Visa . .	102
3.22	Description textuelle du cas d'utilisation Consulter la liste des visas . . . . .	104

# Liste des abréviations

**API** Application Programming Interface

**CIB** Carte Interbancaire

**DZD** Dinar Algérien

**HTML** HyperText Markup Language

**HTTP** Hypertext Transfer Protocol

**JSON** JavaScript Object Notation

**JWT** JSON Web Token

**MVC** Model-View-Controller

**MySQL** My Structured Query Language

**OCR** Optical Character Recognition

**ORM** Object-Relational Mapping

**PDF** Portable Document Format

**PHP** Hypertext Preprocessor

**PK** Primary Key

**SDK** Software Development Kit

**SGBDR** Système de Gestion de Bases de Données Relationnelles

**SQL** Structured Query Language

**UML** Unified Modeling Language

**URL** Uniform Resource Locator

## Liste des mots clés

- Application web
- Gestion de la relation client (CRM)
- Agence de voyage
- Tourisme
- API Graph Facebook
- API Instagram
- Chargily ePay
- Stripe
- API Amadeus
- Paiement en ligne
- Recherche en temps réel
- Scrum
- Modélisation UML
- Développement web
- Webhooks
- Quotas API

# Introduction Générale

À l'ère du numérique, l'informatique et les technologies web jouent un rôle crucial dans la transformation des entreprises, notamment dans le secteur du tourisme. Les agences de voyage s'appuient de plus en plus sur des plateformes digitales performantes pour gérer leurs offres, améliorer leur visibilité et interagir avec leurs clients en temps réel. Le développement d'applications web adaptées constitue ainsi un levier essentiel pour répondre aux attentes d'un marché en constante évolution.

Parmi les tendances les plus marquantes de cette transformation figure l'essor rapide des paiements électroniques en Algérie. À titre d'exemple, le montant des paiements en ligne est passé de 32,3 à 51,9 milliards de dinars entre 2023 et 2024. Les transactions mobiles ont également connu une hausse de 49 Les réseaux sociaux, comme Facebook et Instagram, jouent un rôle central dans les stratégies de communication et marketing des agences touristiques. Ils permettent non seulement de promouvoir les offres, mais aussi d'interagir directement avec les clients et de faciliter les réservations, renforçant ainsi visibilité, confiance et réactivité.

Le présent mémoire s'inscrit dans cette dynamique, à travers la conception et la réalisation d'une application web pour la gestion de l'agence de voyage Ziguade Tour. Cette application permet non seulement de gérer efficacement les offres de voyages (dont les Omras), mais aussi de les publier automatiquement sur plusieurs plateformes (site web, Facebook, Instagram), de centraliser les interactions utilisateurs (likes, commentaires, avis) et de permettre les réservations avec paiement en ligne(national et international). Elle répond ainsi à une double exigence : l'optimisation interne des processus et l'amélioration de l'expérience client. Afin de mener à bien ce projet, nous avons adopté une approche agile basée sur la méthode Scrum, structurée en sprints successifs permettant une planification, une modélisation et une réalisation progressive et itérative du produit.

Le présent mémoire est structuré en quatre chapitres complémentaires, chacun abordant

une étape clé du projet.

- **Chapitre 1 — Cadre du travail et problématique** : présente l'organisme d'accueil, définit les concepts du tourisme, analyse l'existant, identifie la problématique, la solution et établit le cahier des charges fonctionnel et technique.
- **Chapitre 2 — Spécification des besoins** : expose la méthode agile Scrum adoptée, identifie les acteurs, décrit les User Stories, les Backlogs, les outils utilisés ainsi que la planification des sprints.
- **Chapitre 3 — Étude des Sprints 1, 2 et 3** : détaille les fonctionnalités développées durant les trois sprints à travers des cas d'utilisation, des interfaces, des diagrammes UML et des modèles relationnels.
- **Chapitre 4 - Intégration des APIs externes : Réseaux sociaux et Paiement** : dédié à l'intégration des APIs externes, notamment celles de Facebook, Instagram (Meta Graph API) et des solutions de paiement Stripe et Chargily. Il présente les aspects techniques, les contraintes, les tests en environnement sécurisé et la comparaison des solutions utilisées.

# Chapitre 1: Cadre du travail et Problématique

## 1.1 Introduction

Béjaïa, ville côtière du nord-est algérien, est une destination touristique appréciée grâce à sa richesse naturelle, culturelle et historique. Chaque année, elle attire de nombreux visiteurs, aussi bien nationaux qu'internationaux. Face à cette dynamique touristique, les agences de voyage locales comme ZIGUADE TOUR cherchent à moderniser leurs services pour mieux répondre aux besoins des voyageurs.

Avec l'évolution des technologies et les nouvelles attentes des clients, la gestion traditionnelle ne suffit plus. La mise en place d'une solution numérique devient essentielle pour améliorer la qualité des services, faciliter les réservations et gérer efficacement les offres.

Dans ce chapitre, nous allons présenter les concepts clés liés au tourisme, les différents types de tourisme existants, ainsi que l'organisme d'accueil dans lequel s'inscrit notre projet.

## 1.2 Définition et concepts du tourisme

### 1.2.1 Définition du tourisme

Le tourisme désigne les activités des personnes qui voyagent et séjournent en dehors de leur environnement habituel pour une période consécutive qui ne dépasse pas une année, à des fins de loisirs, d'affaires ou d'autres. Ce concept englobe les déplacements, l'hébergement, les services utilisés, ainsi que les interactions sociales et culturelles qui peuvent en découler. [1]

### 1.2.2 Typologie du tourisme

Le tourisme se décline en plusieurs catégories, en fonction des motivations et des attentes des voyageurs. Parmi les types les plus courants, on peut citer [2] :

- **Le tourisme culturel** : voyager pour découvrir l'histoire, l'art et les traditions d'une destination, à travers la visite de musées, monuments et sites historiques.
- **Le tourisme de luxe** : voyager pour bénéficier d'expériences exclusives et haut de gamme, avec des services personnalisés et un confort exceptionnel.
- **Le tourisme urbain** : voyager pour explorer les grandes villes, découvrir leur architecture, leur vie culturelle et leurs événements.
- **Le tourisme balnéaire (tourisme bleu)** : voyager pour profiter des plages, de la mer et des activités nautiques dans des destinations côtières.
- **Le tourisme religieux ou spirituel** : il s'agit de l'ensemble des voyages effectués dans le but de vivre une expérience spirituelle ou de renouer avec sa foi.

## 1.3 Présentation de l'organisme d'accueil

### 1.3.1 Présentation de ZIGUADE TOUR

Ziguade Tour est une agence de voyages et de tourisme située à Béjaïa, en Algérie, précisément à la Cité 208 Logements, Bâtiment A2, Quartier Seghir, en face de la radio Soummam. Elle se positionne comme un partenaire de confiance pour tous types de projets de voyage : qu'ils soient personnels, religieux, touristiques, professionnels ou académiques. Guidée par une équipe engagée et attentive, Ziguade Tour place la satisfaction client au cœur de sa démarche. L'agence offre un large éventail de services adaptés aux besoins de sa clientèle :

- **Voyages personnalisés** : création de voyages sur mesure selon les préférences et le budget du client.
- **Organisation de l'Omra** : prise en charge complète du pèlerinage, avec encadrement.
- **Réservation d'hôtels** : hébergements de qualité grâce à des partenariats avec de nombreux établissements.
- **Transport** : gestion des déplacements en bus, en voiture ou en avion.
- **Assistance visa** : aide pour les demandes de visa touristique, professionnel et d'études à l'étranger.
- **Assurance voyage** : couverture complète pour garantir la sécurité durant les séjours.
- **Assistance à la billetterie** : accompagnement dans la réservation de billets d'avion via les plateformes spécialisées (Air Algérie, Turkish Airlines, Qatar Airways, etc.).

Ziguade Tour travaille en partenariat avec plusieurs compagnies aériennes nationales et internationales telles que : Air Algérie, Tunisair, Turkish Airlines, Qatar Airways, Lufthansa, Air France, Emirates, Alitalia, EgyptAir, entre autres.



FIGURE 1.1 – Logo de l'organisme d'accueil ZIGUADE TOUR

### 1.3.2 Historique de l'agence ZIGUADE TOUR

Fondée en 2013 à Béjaïa, Ziguade Tour est née de la volonté d'offrir aux habitants de la région des solutions de voyage accessibles, fiables et professionnelles. Depuis sa création, l'agence n'a cessé d'évoluer, en développant progressivement son offre de services et en établissant des partenariats solides avec des prestataires locaux et internationaux. Grâce à son expérience, son engagement envers la qualité, et sa relation de proximité avec ses clients, Ziguade Tour s'est imposée comme une référence régionale dans le secteur du voyage et du tourisme.

## 1.4 L'étude de l'existant

Pour concevoir une application efficace, il est essentiel d'analyser les agences de voyage existantes afin d'identifier leurs technologies, services et modes de réservation. Cette étude comparative des agences locales, nationales et internationales met en évidence leurs points forts et leurs limites, permettant ainsi de proposer une solution optimisée et innovante.

### 1.4.1 Description des applications existantes

Dans cette section, nous analysons les agences locales, nationales et internationales en mettant en avant les plateformes numériques qu'elles exploitent pour proposer leurs services.

#### 1.4.1.1 Agences locales à Béjaïa

Béjaïa compte des centaines d'agences (124 agences de voyage selon une recherche google effectuée le 20/02/2025 [3]), mais seules celles mentionnées ci-dessous disposent d'une plateforme web. Les autres agences, comme Béjaïa Tours, Ziguade Tour, Fly Voyages, Tafsut Travel, ainsi que plusieurs autres, se limitent principalement à des comptes sur les réseaux sociaux tels que Facebook, Instagram et WhatsApp pour interagir avec leur clientèle.

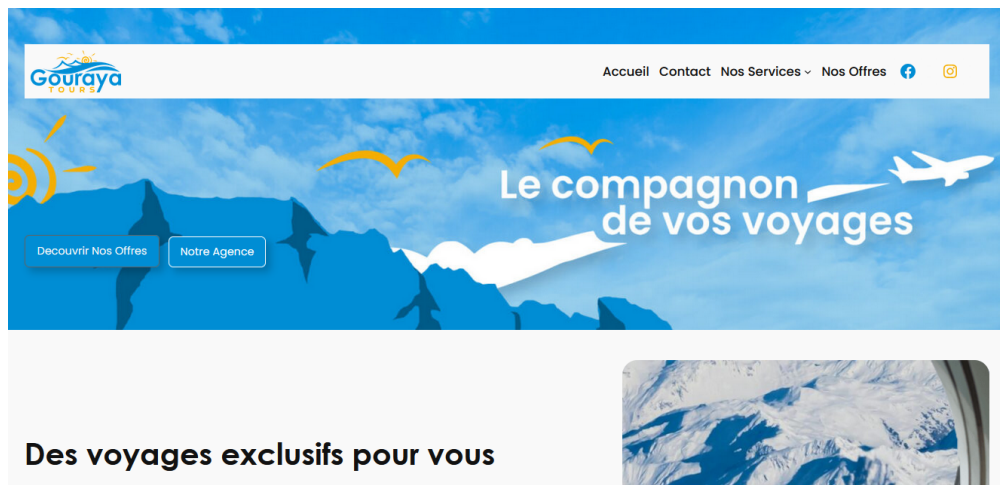


FIGURE 1.2 – Site de Gouraya Tour

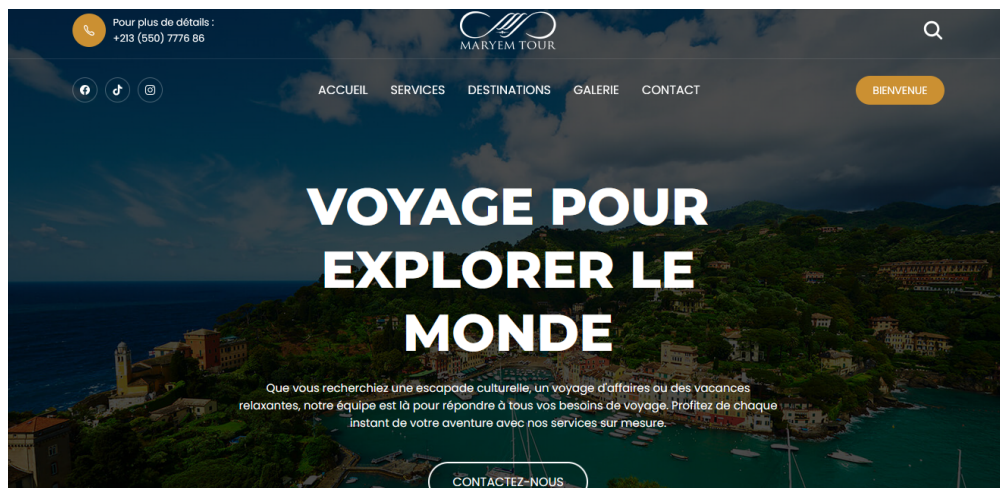


FIGURE 1.3 – Site de Meryam Tour

### 1.4.1.2 Agences nationales en Algérie

Les agences nationales couvrent un territoire plus large et offrent des services variés, incluant les voyages internationaux, les packages touristiques et parfois des solutions en ligne pour la réservation.

#### 1.4.1.2.1 Dune Voyages

Dune Voyages est une application web d'une agence de voyage et du tourisme algérienne, basée à Kouba et fondée en 1996. Cette plateforme interactive facilite la planification des voyages et des séjours, ainsi que la réservation de services tels que les vols, les hôtels et d'autres prestations touristiques.

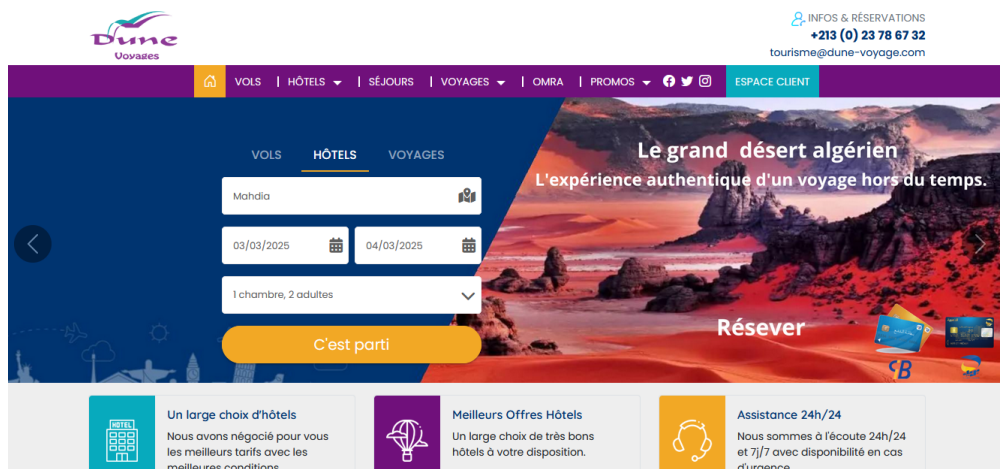


FIGURE 1.4 – Page d'accueil du site web Dune Voyages

#### 1.4.1.2.2 H24voyages

H24 Voyages est une application web d'une agence de voyage basée en Algérie, spécialisée dans la réservation d'hôtels et d'autres services touristiques. Elle permet aux utilisateurs de consulter les offres disponibles et de réserver directement via une interface intuitive. Le site propose un système de paiement en ligne sécurisé, bien que le nombre d'établissements et de destinations reste limité.

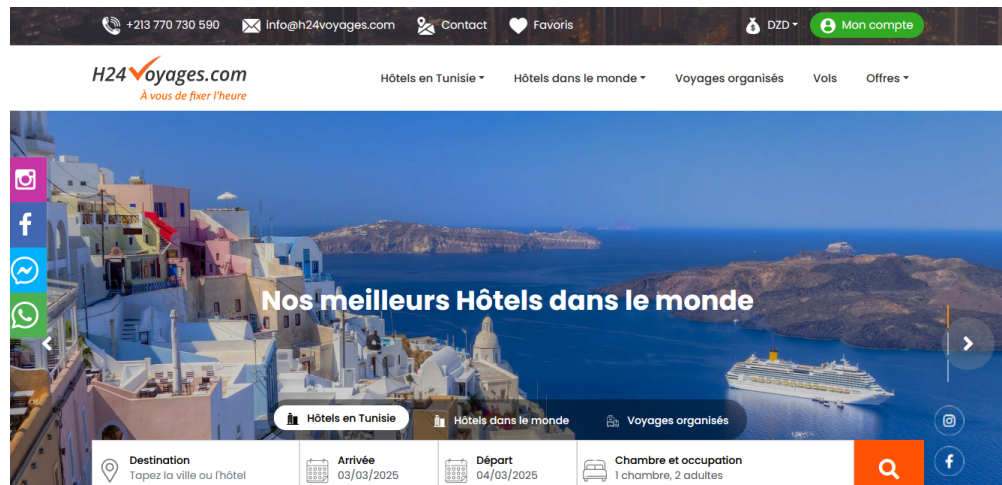


FIGURE 1.5 – Page d'accueil du site web H24Voyages

### 1.4.1.3 Agences internationales

Les agences internationales disposent d'une présence mondiale et utilisent des plateformes avancées pour la réservation et la gestion des voyages.

#### 1.4.1.3.1 Fram

est une application web d'une agence de voyages française avec plus de 70 ans d'expérience, spécialisée dans la réservation de vols et d'hôtels. Elle offre une variété de séjours, de services et de destinations exceptionnelles, ainsi qu'un service client dédié pour accompagner les voyageurs tout au long de leur expérience.

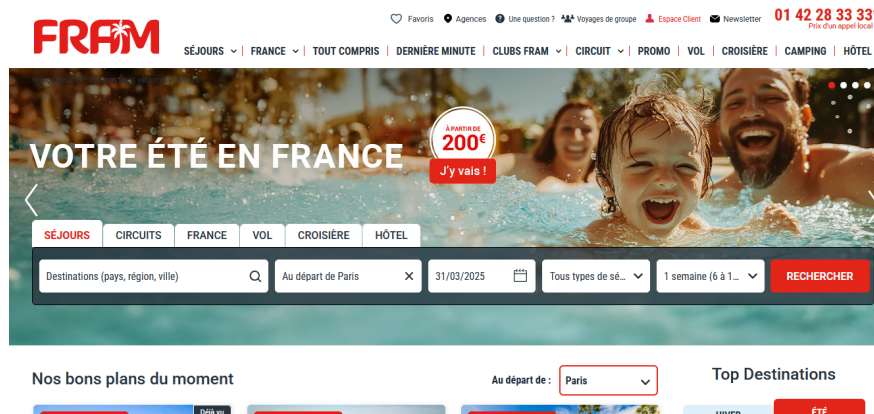


FIGURE 1.6 – Page d'accueil du site web Fram

### 1.4.1.3.2 Booking

Booking est une application web spécialisée dans la réservation d'hôtels, de vols, et d'autres services de voyage. Elle permet aux utilisateurs de rechercher, réserver et payer des hébergements, tout en offrant des informations détaillées et des avis clients. Grâce à son API, elle facilite l'intégration de ces services dans des applications externes.

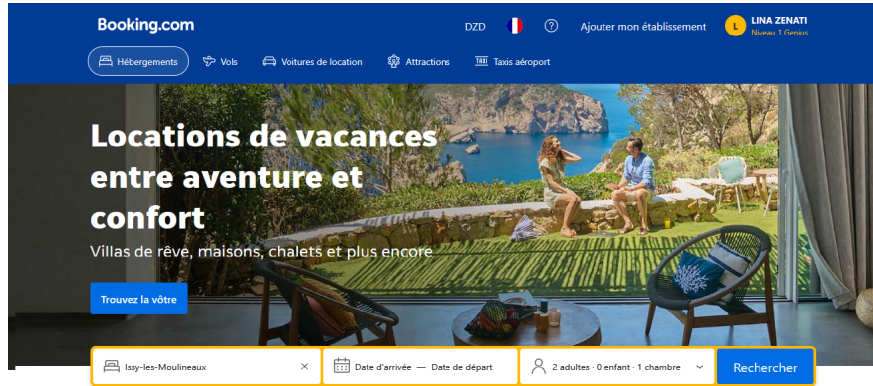


FIGURE 1.7 – Page d'accueil du site web Booking

## 1.4.2 La comparaison des services et technologies des applications web

Dans cette section, nous comparons les applications web des agences de voyage en fonction des technologies qu'ils utilisent, de leurs performances et de leurs services en ligne. Cette analyse permet de mieux comprendre les choix techniques, l'accessibilité et l'efficacité des plateformes.

### 1.4.2.1 Les Technologies

Les applications web d'agences de voyage emploient une variété de technologies pour assurer le bon fonctionnement de leurs services. Pour identifier ces outils, nous avons utilisé l'extension Wappalyzer. Wappalyzer est un outil disponible sous forme d'extension de navigateur qui liste l'ensemble des technologies employées sur une application web[4]

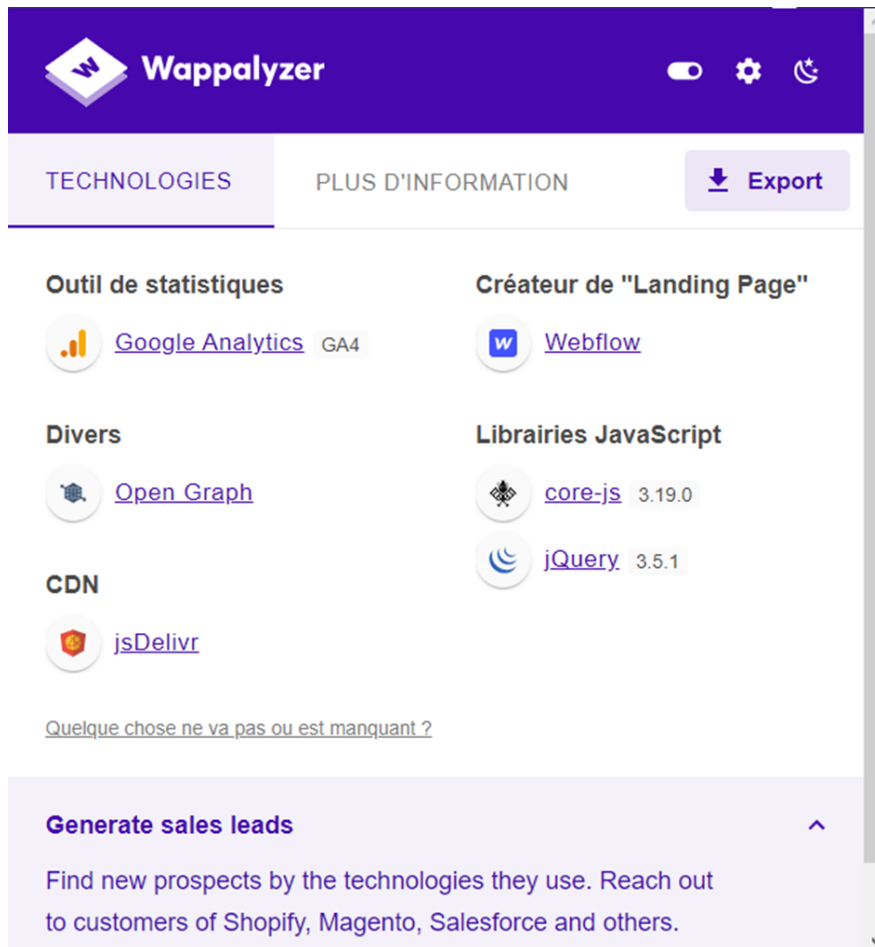


FIGURE 1.8 – Logo de l’outil Wappalyzer

Selon les données recueillies via Wappalyzer, nous avons analysé les frameworks, langages de programmation, CDN et autres outils utilisés sur ces plateformes. Le tableau 1.1 résume ces informations :

Critères	Gouraya Tours	Maryem Tour	Dune Voyages	H24voyages	Fram	Booking
Type de site web	Dynamique	Statique	Dynamique	Dynamique	Dynamique	Dynamique
Langages & Frameworks	animate.css, PHP, MySQL, HTML, CSS, JS, Bootstrap	HTTP/2, HTML, CSS	Bootstrap, animate.css, HTTP/2, HTML, CSS, JS	jQuery, Moment.js, Typeahead.js, LazySizes, SweetAlert2	Bootstrap, React, HTTP/2, HTML, CSS, JS	Onsen UI, HTML, CSS, JS
Librairies JS	jQuery, jQuery Migrate, Slick, core-js	aucun	JavaScript, jQuery, Moment.js, SweetAlert, Typeahead.js, LazySizes	cdnjs, jsDelivr, jQuery CDN	Core-js	PubSubJS, Lodash, jQuery Migrate, jQuery
CDN	aucun	Unpkg, cdnjs, Cloudflare	cdnjs, jsDelivr	Unpkg, Google Hosted Libraries, Cloudflare	Google Font API, Amazon CloudFront	Amazon CloudFront
APIs externes	Aucune	Google Fonts API	Mapbox, Google Maps, Google Flights, API de paiement, API Booking	API de paiement, API Booking	API Booking, API compagnies aériennes, API paiement	API hôtels, API compagnies aériennes

TABLE 1.1 – Comparaison des technologies des applications web

### 1.4.2.2 Analyse critique :

Les agences locales comme Gouraya Tours ou Maryem Tour utilisent des technologies simples (HTML, CSS, PHP, MySQL) dans leurs applications web, ce qui peut limiter la performance, la sécurité et l'expérience utilisateur. En revanche, les plateformes internationales comme Booking ou Fram utilisent des technologies plus avancées (React, Node.js, services cloud, API externes), leur permettant d'offrir des sites plus rapides, sécurisés et évolutifs. Ces écarts technologiques justifient notre choix d'utiliser des outils modernes dans notre propre application.

### 1.4.3 L'analyse des performances

Pour évaluer les performances techniques des sites web, nous avons utilisé la plateforme Site Analyzer.

Le site [www.site-analyzer.com](http://www.site-analyzer.com) est proposé par la Société VSI INNOVATION, il permet à toute personne physique ou morale d'analyser des sites Internet afin d'avoir accès à un rapport multicritères détaillé. Il permet également d'analyser la qualité des liens pointant vers les sites, le suivi de position Google et la comparaison avec les concurrents.[5]

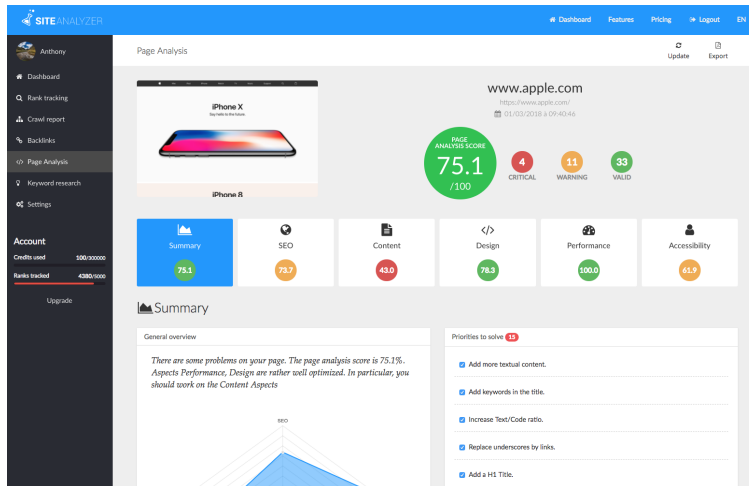


FIGURE 1.9 – page d'accueil du site web site-analyzer

À l'aide de Site Analyzer, nous avons évalué plusieurs critères techniques liés aux performances des sites, notamment leur taille, leur temps de chargement et leur score de performance. Voilà le tableau 1.2 détaillant ces aspects :

Critères	Gouraya Tours	Maryem Tour	Dune Voyages	H24voyages	Fram	Booking
Taille du fichier (kb)	45.99	4.58	11.8	20.9	65.74	il est impossible d'obtenir les résultats
Taux de compression	79 %	80 %	84 %	79%	84 %	il est impossible d'obtenir les résultats
Temps d'exécution (ms)	2	82	98	176	4	il est impossible d'obtenir les résultats
Score de performance	C	A	B	B	C	il est impossible d'obtenir les résultats
Accessibilité	La taille de nom de domaine est bonne	La taille de nom de domaine est trop longue	La taille de nom de domaine est bonne	La taille de nom de domaine est bonne	La taille de nom de domaine est bonne	il est impossible d'obtenir les résultats
Usabilité	Une seule Balise Noscript a été définie, La Balise Bgsound est manquante	Les Balises Noscript et Bgsound sont manquantes	Les Balises Noscript et Bgsound sont manquantes	Une seule Balise Noscript a été définie, La Balise Bgsound est manquante	Une seule Balise Noscript a été définie, La Balise Bgsound est manquante	il est impossible d'obtenir les résultats
Internationalisation	Français	Français	Français	Français, Anglais	45 Langues	45 Langues
Partage	Il n'utilise pas Le protocole open graph, Il n'utilise pas Twitter card	Il n'utilise pas Le protocole open graph, Il n'utilise pas Twitter card	Il n'utilise pas Le protocole open graph, Il n'utilise pas Twitter card	Utilise Le protocole open graph, Il n'utilise pas Twitter card	Utilise Le protocole open graph, Il n'utilise pas Twitter card	Il n'utilise pas Twitter card
Compatibilité mobile	Responsive	Responsive	Responsive	Responsive	Responsive	Responsive
Doctype	HTML5	HTML5	HTML5	HTML5	HTML5	il est impossible d'obtenir les résultats
Encodage	Header & meta : utf-8	Header & meta : utf-8	Header & meta : utf-8	Header & meta : utf-8	Header & meta : utf-8	il est impossible d'obtenir les résultats
SSL	Activé	Activé	Activé	Activé	Activé	
Serveur IP	91.121.51.179	64.29.17.129	46.105.57.219	46.105.57.219	195.149.66.20	il est impossible d'obtenir les résultats
Qualité de la conception	Activé	Activé	Activé	Activé	Activé	il est impossible d'obtenir les résultats
Sécurité	Faible	Faible	Faible	Faible	Faible	il est impossible d'obtenir les résultats
Score du Site	66.6 %	53.5 %	64.2 %	74.5 %	77 %	il est impossible d'obtenir les résultats

TABLE 1.2 – comparaison des applications web

#### **1.4.3.1 Analyse critique :**

Ce tableau montre deux types de scores : le score de performance (lié à la taille du fichier, au taux de compression, au temps d'exécution et à la qualité du code) et le score global du site (qui prend aussi en compte la sécurité, l'accessibilité, l'internationalisation, etc.).

Par exemple, l'application web de Maryem Tour a un bon score de performance (A) grâce au respect des standards, malgré un temps d'exécution élevé. À l'inverse, l'application web de Gouraya Tours est très rapide (2 ms) mais obtient un score de performance faible (C) à cause d'un code moins optimisé.

Les plateformes comme l'application web de Fram et celle de l'agence locale H24voyages obtiennent un bon score global (77). Cela justifie notre choix de développer une application à la fois rapide, bien structurée et conforme aux bonnes pratiques du web.

#### **1.4.4 Modes de Réservation et de Paiement**

Les sites web des agences offrent différentes approches pour la réservation et le paiement des services. Certains permettent une réservation et un paiement entièrement en ligne, tandis que d'autres nécessitent un contact avec l'agence. Le tableau 1.3 présente une comparaison des modes de réservation et de paiement disponibles sur ces plateformes

Critères	Gouraya Tours	Maryem Tour	Dune Voyages	H24voyages	Fram	Booking
Mode de Réservation des hôtels	En agence (Formulaire de contact)	En agence, Contact par email, téléphone	Formulaire de Réservation envoyé à l'agence qui va la gérer	En agence, Contact par email, téléphone	En ligne (En temps réel), Réservation immédiate envoyé aux hôtels directs via une api externe	En ligne (En temps réel), Réservation immédiate envoyé aux hôtels directs via une api externe
Mode de Réservation des vols	En agence, Formulaire de contact	En agence, Contact par email, téléphone	En ligne (En temps réel), Réservation immédiate envoyé aux compagnies directes via une api externe	Formulaire de contact, En Agence	En ligne (En temps réel), Réservation immédiate envoyé aux compagnies directes via une api externe	En ligne (En temps réel), Réservation immédiate envoyé aux compagnies directes via une api externe
Mode de paiement	En agence	En agence	En ligne, en agence	En agence, en ligne	En ligne	En ligne
Mode de Réservation de Voyage Organisé	En agence	En agence	En agence	Formulaire de Réservation envoyé à l'agence qui va la gérer	En ligne	En ligne
Mode de Filtrage	- Vols : pas disponible, - Hôtels : pas disponible	- Vols : pas disponible, - Hôtels : pas en temps réel	- Vols : en temps réel, - Hôtels : pas en temps réel	- Vols : pas disponible, - Hôtels : pas en temps réel	- Vols : en temps réel, - Hôtels : en temps réel	- Vols : en temps réel, - Hôtels : en temps réel

TABLE 1.3 – comparaison des modes de réservation et de filtrage des applications web

#### 1.4.4.1 Analyse Critique

Ce tableau montre que les agences locales comme Gouraya Tours ou Maryem Tour restent très dépendantes des réservations et paiements en agence, ce qui limite leur accessibilité et leur réactivité face aux besoins des clients. À l'inverse, des plateformes comme Booking et Fram offrent des services en ligne complets, avec réservation et paiement immédiats, ainsi qu'un filtrage en temps réel qui facilite grandement la recherche et la prise de décision. Cette différence met en évidence un retard technologique important des agences locales, qui pourraient améliorer leur compétitivité en adoptant des solutions digitales plus modernes et en intégrant des systèmes automatisés de réservation et paiement en ligne.

#### 1.4.5 Services et offres

Cette section présente une analyse comparative des services et offres disponibles sur les sites web des agences de voyage étudiées. Chaque plateforme propose un ensemble de fonctionnalités liées à la billetterie, à l'hébergement, au transport, ainsi qu'à différents types de séjours et d'expériences touristiques. Le tableau 1.4 synthétise les principales différences observées.

Critères	Gouraya Tours	Maryem Tour	Dune Voyages	H24voyages	Fram	Booking
Billetterie	Vols non disponibles en ligne, uniquement réservables en agence	Vols non disponibles en ligne, uniquement réservables en agence	Vols nationaux et internationaux disponibles en ligne	Vols non disponibles en ligne, uniquement réservables en agence	Vols nationaux et internationaux disponibles en ligne	Vols nationaux et internationaux disponibles en ligne
Hébergement	Hébergement non disponible en ligne, uniquement en agence	Hébergement non disponible en ligne, uniquement en agence	Hôtels, appartements, maisons de vacances	Consultation des hébergements possible sans fonctionnalité de réservation en ligne	Hôtels, appartements, maisons de vacances	Hôtels, appartements, maisons de vacances
Transport	Descriptions disponibles uniquement, sans possibilité de réservation	Descriptions disponibles uniquement, sans possibilité de réservation	Non proposé	Service non disponible	Croisières proposées	Location de voitures, taxis aéroport
Types de séjours et expériences	Absence de séjours ou expériences interactives sur le site	Absence de séjours ou expériences interactives sur le site	Tourisme organisé en groupe, tourisme religieux (Omra, Hajj)	Tourisme organisé en groupe, tourisme individuel et sur mesure	Tourisme organisé en groupe, tourisme individuel et sur mesure	Tourisme individuel et sur mesure, tourisme culturel et de loisirs
Langues	Français uniquement	Français uniquement	Français uniquement	Français uniquement	45 langues disponibles	45 langues disponibles
Devises	DZD uniquement	DZD uniquement	DZD uniquement	DZD uniquement	51 devises prises en charge	54 devises prises en charge
Public cible	Clients algériens	Clients algériens	Clients algériens et internationaux	Clients algériens	Clients algériens et internationaux	Clients algériens et internationaux
Service client	Formulaire de contact en ligne	Contact par e-mail et téléphone	Contact par e-mail et téléphone	Contact par e-mail, téléphone et newsletter	Contact par téléphone et newsletter	Contact par e-mail, téléphone, questionnaire et newsletter
Liste des pays et villes concernées	Liste non disponible en ligne, nécessite un contact avec l'agence	Liste non disponible en ligne, nécessite un contact avec l'agence	Liste fixe (Algérie, Égypte, Émirats Arabes Unis, Malaisie, Maldives)	Liste fixe (Tunisie, Turquie, Paris, Égypte)	Liste ouverte	Liste ouverte
Présence sur les réseaux sociaux	Facebook, Instagram	Facebook, TikTok	Facebook, Instagram, Twitter	Facebook, Instagram, WhatsApp	Facebook, Instagram, TikTok, YouTube, Twitter, Pinterest, blog	Facebook, Instagram, TikTok, LinkedIn, Pinterest, blog

TABLE 1.4 – Comparaison des services des applications web

### 1.4.5.1 Analyse Critique

Ce tableau révèle que les agences locales comme Gouraya Tours et Maryem Tour proposent des services très limités, souvent accessibles uniquement en agence, ce qui restreint leur attractivité et leur portée. À l'inverse, des plateformes comme Booking et Fram offrent une large gamme de services en ligne, couvrant la billetterie, l'hébergement, le transport et des expériences touristiques diversifiées, avec une présence multilingue et une adaptation aux devises internationales. Cette différence illustre une forte disparité en termes d'accessibilité, de diversité d'offres et d'orientation client, soulignant la nécessité pour les agences locales

d'adopter davantage la digitalisation pour rester compétitives dans un marché globalisé.

### 1.4.6 Critique de l'existant

Dans cette section, Nous étudions les avantages et inconvénients des applications citées précédemment afin d'évaluer leur efficacité et leurs limites. Le Tableau 1.5 met en évidence les forces et faiblesses de chaque plateforme.

Sites web	Avantages	Inconvénients
Gouraya Tours et Maryem Tour	<ul style="list-style-type: none"> <li>- Aucune dépendance à une connexion internet</li> <li>- Possibilité de négociation des prix</li> <li>- Possibilité de paiements en plusieurs fois</li> </ul>	<ul style="list-style-type: none"> <li>- Dépendance aux réseaux sociaux</li> <li>- Pas d'interaction en temps réel</li> <li>- Expérience utilisateur limitée à des pages fixes</li> </ul>
Dune Voyages	<ul style="list-style-type: none"> <li>- Large choix de destinations et d'offres</li> <li>- La réservation des vols en temps réel</li> <li>- Système de paiement en ligne intégré</li> </ul>	<ul style="list-style-type: none"> <li>- Informations limitées sur la disponibilité des hôtels</li> <li>- Confirmation de réservation non instantanée</li> <li>- Absence de service client en ligne</li> </ul>
H24voyages	<ul style="list-style-type: none"> <li>- Interface simple et claire</li> <li>- Plusieurs méthodes de paiement disponibles en ligne/en agence</li> <li>- Comparaison des prix directement sur le site</li> </ul>	<ul style="list-style-type: none"> <li>- Obligation de paiement total à l'avance</li> <li>- Absence de support client</li> <li>- Offres et services limités</li> </ul>
Fram	<ul style="list-style-type: none"> <li>- Fiabilité et confiance</li> <li>- Suivi de réservation en ligne</li> <li>- Service client accessible</li> </ul>	<ul style="list-style-type: none"> <li>- Mauvaise gestion des imprévus</li> <li>- Informations marketing trompeuses</li> <li>- Espace client difficile d'accès</li> </ul>
Booking	<ul style="list-style-type: none"> <li>- Grande visibilité mondiale</li> <li>- Large choix d'hébergements et filtres avancés</li> <li>- Réservation instantanée et confirmation rapide</li> </ul>	<ul style="list-style-type: none"> <li>- Absence de garantie de la part de Booking</li> <li>- Risque en cas d'arnaque</li> <li>- Politique de remboursement incertaine</li> </ul>

TABLE 1.5 – Avantages et inconvénients des sites web de réservation

## 1.5 Problématique et objectifs du travail

### 1.5.1 Problématique

Le secteur du tourisme en Algérie connaît une croissance notable ces dernières années, avec une hausse du nombre de voyageurs domestiques et internationaux. En 2023, le pays a enregistré plus de 3,2 millions de touristes, soit une augmentation de 135,5 % par rapport à 2022. Le nombre de touristes étrangers entrants a, quant à lui, augmenté de 66,68 %, atteignant 2 195 519 visiteurs. Concernant le tourisme intérieur, la saison saharienne 2023–2024 a attiré plus de 254 000 touristes entre octobre et décembre, dont la majorité étaient des résidents algériens (Ministère du Tourisme et de l'Artisanat, MTA, 2023).

Lorsqu'un site web existe, il se limite souvent à une vitrine statique, sans fonctionnalités interactives ni services connectés intelligents.

Parmi les principales limites observées, on note :

- Le manque de possibilité de paiement en ligne, que ce soit via la carte Dahabia, CIB ou une carte bancaire internationale. Les clients doivent souvent se rendre physiquement

à l'agence pour confirmer une réservation, ce qui est contraignant, notamment pour les clients vivant loin ou à l'étranger.

- L'absence d'intégration avec des API temps réel pour les vols et les hôtels, empêchant les utilisateurs de consulter des offres actualisées, de vérifier les disponibilités ou de comparer les prix instantanément.
- Le non-recours aux outils modernes de communication et de marketing, comme l'API Graph de Facebook ou l'API Instagram. Cette carence se traduit par :
  - L'absence de publications régulières et automatisées.
  - Des commentaires laissés sans réponse, même s'ils sont positifs, ce qui nuit à l'image de l'agence.
  - Une difficulté à voir facilement ce que le public pense ou fait (likes, partages, avis), et à répondre rapidement à leurs réactions.
  - Une difficulté à contrôler les messages irrespectueux ou inappropriés que certains utilisateurs publient.
- Une absence de centralisation des services, comme les demandes de visa, d'assurance ou les réservations de vols et d'hôtels. Ces démarches sont souvent traitées séparément, manuellement et sans suivi numérique, ce qui allonge les délais et augmente le risque d'erreurs.

Tous ces éléments contribuent à une expérience utilisateur fragmentée, inefficace et peu attractive, rendant les agences locales moins compétitives par rapport aux grandes plateformes internationales.

## 1.5.2 Solution proposée

L'application **ZIGUADE TOUR** propose une solution complète, moderne et intelligente, qui répond à l'ensemble de ces problématiques à travers une plateforme centralisée et modulaire, conçue pour les besoins réels du secteur.

Les points forts de la solution proposée incluent :

- La possibilité pour les utilisateurs inscrits et clients de réaliser un paiement en ligne sécurisé, intégré directement dans le processus de réservation des hôtels, vols, Omra et voyages organisés, via carte CIB, Dahabia ou carte bancaire internationale (Visa/Mastercard), grâce à l'intégration des APIs de paiement **Chargily ePay** pour les paiements nationaux et **Stripe** pour les paiements internationaux.
- La possibilité pour l'administrateur de configurer des APIs d'hôtels et de vols, notamment :
  - L'API Amadeus pour l'accès en temps réel aux offres de vols (disponibilités, tarifs, horaires).
  - Une API hôtelière interne permettant l'affichage instantané des hébergements disponibles, sans passer par des services payants tiers.
- La possibilité pour l'administrateur de configurer l'API Graph de Facebook et Instagram, permettant :

- La publication automatisée des offres (Omra et voyages organisés) sur les réseaux sociaux.
- L’affichage dynamique des avis clients et du nombre de likes sur chaque offre.
- La possibilité de masquer automatiquement les messages irrespectueux, grâce aux paramètres de modération de l’API.
- La gestion facilitée de la présence numérique sur les réseaux sociaux.
- Un système de réservation unifié pour tous les services : hôtels, vols, voyages organisés, Omra, visas, assurances.
- Un tableau de bord personnalisé est disponible pour chaque type d’utilisateur :
  - Le visiteur, l’utilisateur inscrit et le client ont accès à un tableau de bord *FrontOffice* pour consulter, réserver et payer.
  - L’agent et l’administrateur partagent le même tableau de bord *BackOffice*, avec des fonctionnalités limitées pour l’agent et des capacités étendues pour l’administrateur, telles que la configuration des APIs.

## 1.6 Cahier des charges

### 1.6.1 Besoins fonctionnels

Acteur	Fonctionnalités
	<p><b>1. Gérer mon compte personnel</b></p> <ul style="list-style-type: none"> <li>- modifier mes informations personnels</li> </ul>
	<p><b>2. Gérer les Voyages organisés &amp; Omras</b></p> <ul style="list-style-type: none"> <li>- Ajouter, modifier et supprimer les voyages / omras qui sont pas publiés</li> <li>- modifier les voyages / omras qui sont publié et date de fin n’est pas dépassé</li> <li>- publier le même voyage organisé ou Omra sur site /facebook /instagram</li> <li>- Recevoir et consulter les demandes de réservation soumises par les utilisateurs inscrits</li> <li>- Accepter(autoriser le paiement) ou refuser la demande de réservation.</li> <li>- annuler ou modifier la demande de réservation des Voyages organisés Omras</li> </ul>

### **3. Gérer les publications partagés sur les réseaux sociaux (facebook/instagram)**

- récupérer les publications
- affichage du nombre restant de publications possibles
- régénérer les nouveaux commentaires de chaque publication
- affichage du nombre restant de récupération des commentaires et suppression possibles
- récupérer les anciens commentaires de chaque publication
- partager les commentaires récupérer sur le site en tant que des avis
- supprimer les commentaires de chaque publication
- récupérer le nombre de likes de chaque publication

### **4. Gérer les assurances**

- Recevoir et consulter les demandes d'assurance soumises par les utilisateurs inscrits.
- Accepter (autoriser le paiement) ou refuser une demande d'assurance
- supprimer une demande d'assurance

### **5. Gérer des visas (touristique, études, affaire)**

- Recevoir et consulter les demandes de visas soumises par les utilisateurs inscrits.
- Accepter (autoriser le paiement) ou refuser une demande de visa
- supprimer une demande de visa

### **6. Gérer les vols & Hôtels**

- Ajouter une nouvelle ville pour explorer ses hôtels
- Recevoir et consulter les demandes de réservation soumises par les utilisateurs inscrits
- Accepter (autoriser le paiement) ou refuser la demande de réservation.
- annuler ou modifier la réservation d'hôtel/vol

### **8. Messagerie**

- Répondre aux messages reçus par les utilisateurs inscrits

<b>Admin</b>	<b>1. Gérer toutes les fonctionnalités accessibles à l'agent.</b>
	<b>2. Gérer les utilisateurs</b> <ul style="list-style-type: none"> <li>- Création, modification et suppression des comptes agents</li> <li>- Suppression des comptes utilisateurs inscrits si ils ont pas connectés plus de deux ans</li> <li>- Recherche et filtrage de la liste des utilisateurs</li> </ul>
	<b>3. Gérer les paiements</b> <ul style="list-style-type: none"> <li>- Consultation et filtrage par statut et date</li> </ul>
	<b>4. Gérer la configuration</b> <ul style="list-style-type: none"> <li>- Modification des clés API (paiement, réseaux sociaux)</li> </ul>
<b>Utilisateur inscrit</b>	<b>1. Créer un compte (inscription)</b> <b>2. Authentification</b>
	<b>2. Gérer mon compte</b> <ul style="list-style-type: none"> <li>- modifier les informations personnels</li> <li>- suivre les messages reçus par l'admin /agent</li> <li>- suivre les réservations(Omra/Voyage/Vol/Hôtel)</li> <li>- suivre les demandes de visa (si la demande est accepté ou non)</li> <li>- suivre les demandes d'assurance (si la demande est accepté ou non)</li> <li>- télécharger les factures</li> </ul>
	<b>3. Vols &amp; hôtels</b> <ul style="list-style-type: none"> <li>- Rechercher et filtrer les vols en temps réel</li> <li>- voir l'emplacement de chaque hôtel a travers la carte maps</li> <li>- Remplir un formulaire de réservation</li> <li>- Recevoir la confirmation de l'admin/agent pour que le paiement soit autorisé</li> <li>- Payer l'acompte en ligne (chargily,stripe)</li> </ul>
	<b>4. Voyages organisés et Omras</b> <ul style="list-style-type: none"> <li>- Consulter les publications (voyages organisés et Omras) du site</li> </ul>

	<ul style="list-style-type: none"> <li>- voir les commentaires de chaque publication ( voyage/omra ) et pouvoir accéder direct a la meme publication sur facebook/instagram</li> <li>- Consulter les voyages organisés et Omras</li> <li>- Remplir un formulaire avec ses coordonnées</li> <li>- Recevoir la confirmation de l’admin/agent pour que le paiement soit autorisé ou non</li> <li>- Payer l’assurance en ligne</li> </ul>
	<p><b>5. Souscription à l’assurance voyage</b></p> <ul style="list-style-type: none"> <li>- Remplir un formulaire avec ses coordonnées</li> <li>- Recevoir la confirmation de l’admin/agent pour que le paiement soit autorisé ou non</li> <li>- Payer l’assurance en ligne</li> </ul>
	<p><b>6. Visa</b></p> <ul style="list-style-type: none"> <li>- Remplir un formulaire de demande de visa (touristique, études, affaires)</li> <li>- Recevoir la confirmation de l’admin/agent pour que le paiement soit autorisé ou non</li> <li>- Payer les frais de visa</li> </ul>
	<p><b>7. Messagerie</b></p> <ul style="list-style-type: none"> <li>- Contacter l’agence via email, numéro de téléphone, Messenger , formulaire de contact.</li> </ul>
	<p><b>8. voir l’emplacement de l’agence a travers la carte Maps</b></p>
<b>Client</b>	<p><b>1. Consulter et gérer le panier</b></p> <ul style="list-style-type: none"> <li>- recevoir les factures de paiements et les télécharger</li> </ul>
	<p><b>2. Recevoir les factures de paiement par email</b></p>
<b>Visiteur</b>	<p><b>1. Inscription</b></p>
	<p><b>2. Visiter le site</b></p>
	<p><b>3. Accéder depuis une publication (voyage/omra) sur Facebook ou Instagram vers la même publication sur le site</b></p>

## 1.6.2 Besoins non fonctionnels

Les besoins non fonctionnels définissent les critères de qualité que l'application doit respecter pour garantir son efficacité, sa fiabilité et sa facilité d'utilisation. Ils concernent notamment la sécurité, la performance, la maintenabilité et l'accessibilité de l'outil.

**Sécurité :** L'accès à l'application doit être protégé par une authentification robuste, et les données personnelles doivent garantir une confidentialité absolue. Voici les mesures détaillées pour répondre à ces exigences :

- L'administrateur ne peut ni modifier ni supprimer un client.
- L'administrateur ne peut pas modifier un utilisateur inscrit, et ne peut le supprimer que si sa dernière connexion remonte à plus de deux ans.
- L'administrateur ne peut pas être supprimé.
- Aucune modification d'un voyage/omra publier sauf si sa date de depart n'est pas encore arrivée.
- Aucune suppression d'un voyage/omra publié.
- il ne faut pas publier le meme voyage/omra sur la meme plateforme .
- Le filtre par plateforme ne doit être disponible que si le filtre "publier" est activé, afin d'éviter la sélection d'une plateforme pour des voyages non publiés.
- On ne peut pas accéder au voyage/omra publié sur site depuis un url de son publication sur une facebook ou instagram ,si mon site il n'est pas encore déployé ou ou si ngrok n'est pas activé.
- Url des publications instagram pour accéder au voyages/omras dans le site n'est pas cliquable contrairement au facebook.

**Performance :** L'application doit répondre rapidement aux actions des utilisateurs, même en cas de forte charge.

**Fiabilité :** Le système doit fonctionner sans erreur et garantir la sauvegarde des données importantes.

**Ergonomie et simplicité :** L'interface doit être claire, facile à comprendre et agréable à utiliser dès la première visite.

**Modularité et maintenabilité :** Le code doit être organisé et facile à modifier ou à améliorer si besoin.

**Disponibilité :** L'application doit être accessible à tout moment, sauf lors des maintenances prévues.

**Accessibilité :** L'application doit bien fonctionner sur tous types d'appareils (ordinateur, tablette, smartphone) et navigateurs.

## 1.7 Conclusion

Dans ce chapitre, nous avons présenté le contexte du tourisme en Algérie et défini les concepts clés du domaine. À travers une étude comparative détaillée des plateformes locales, nationales et internationales, nous avons identifié les principales limites du secteur, ce qui nous a permis de dégager une problématique claire et de fixer les objectifs de notre système ZIGUADE TOUR. Le chapitre suivant sera consacré à la conception de cette solution.

# Chapitre 2: Spécification des besoin

## 2.1 Introduction

Ce chapitre présente les besoins essentiels pour la conception et le développement de notre application. Il permet de mieux comprendre les attentes des utilisateurs et de s'assurer que la solution proposée y répond de manière claire et adaptée. Cette étape est cruciale pour garantir la réussite du projet, car elle sert de base à toutes les phases suivantes du développement.

Nous allons également identifier les différents acteurs impliqués, ainsi que leurs rôles et interactions avec le système, afin d'avoir une vision globale et cohérente des exigences fonctionnelles.

## 2.2 Méthodologie et Conception

Pour structurer et organiser le développement de notre application, nous avons adopté la méthode agile **Scrum**[6], qui repose sur une approche itérative et collaborative. Elle nous a permis de planifier efficacement les différentes phases du projet à travers des sprints, en facilitant la gestion des tâches, le suivi de l'avancement et l'adaptation continue aux besoins. Concernant la modélisation, nous avons utilisé le langage **UML (Unified Modeling Language)**[7], un langage graphique standard permettant de représenter de manière claire et structurée les besoins fonctionnels ainsi que l'architecture du système. Dans notre projet, nous avons principalement utilisé les diagrammes de cas d'utilisation, de séquence et de classes pour illustrer les différentes interactions, fonctionnalités et structures du système.

## 2.3 Identification des acteurs

Notre plateforme intègre plusieurs types d'utilisateurs, chacun disposant de droits d'accès et de fonctionnalités spécifiques. La répartition des rôles s'articule comme suit 2.1

Acteur	Actions
Visiteur	Accède à l'application sans inscription, consulte les offres publiques (voyages, Omra).
Utilisateur inscrit	Crée un compte pour accéder à plus de services : réserver un hôtel, un vol, demander une visa , une assurance et payer en dinars ou en devises, gérer son profil, etc.
Client	Suit ses réservations, consulte son panier et télécharge ses documents personnels.
Agent	Membre du personnel ZIGUADE TOUR chargé de gérer les offres (Omra, voyages), publier sur le site et sur les réseaux sociaux (Facebook, Instagram), gérer les commentaires (récupérer et supprimer), récupérer le nombre de réactions (likes), suivre les réservations(omra,voyage,hotel,vol) et les demandes d'assurances et de visa.
Administrateur	il dispose des mêmes privilèges que l'agent, avec en plus la gestion des utilisateurs, la consultation de la liste des paiements, ainsi que la configuration des API externes (réseaux sociaux, paiements)

TABLE 2.1 – Rôles et actions des différents acteurs de l'application

## 2.4 Gestion de projet avec scrum

### 2.4.1 Les rôles

Pour notre projet, les rôles sont répartis comme indiqué dans le tableau suivant 2.2 :

Rôle Scrum	Personne affectée
Scrum Master	Mr Sider Abderrahmane
Product Owner	Mr Mancer Faycel
Équipe de développement	Zenati Lina et Tiab Mélissa Smicha

TABLE 2.2 – Répartition des rôles Scrum pour le projet

### 2.4.2 Définition des User story

Dans cette section, nous présentons les scénarios d'utilisation typiques du point de vue des différents utilisateurs. Chaque user story décrit une interaction entre un acteur et le système, dans un langage simple et centré sur les objectifs métiers.

## Agent de voyage

En tant qu'agent, je Souhaite Gérer mon compte personnel et modifier mes informations

En tant qu'agent, je Souhaite Gérer un Voyage , Omra (Ajouter , Modifier, Supprimer)

En tant qu'agent, je souhaite publier des offres de voyages organisés ou Omra sur Site, facebook , instagram afin de promouvoir les services de l'agence.

En tant qu'agent je Souhaite Récupérer le nombre de likes d'une publication

En tant qu'agent, je souhaite être notifié des nouveaux commentaires afin de pouvoir les supprimer, et ainsi préserver l'image de l'agence.

En tant qu'agent, je souhaite voir les publication de chaque voyage/omra

En tant qu'agent, je souhaite voir le nombre de qouta restant pour pouvoir publier sur les réseaux sociaux

En tant qu'agent, je souhaite voir les anciens commentaires de chaque publication

En tant qu'agent, je souhaite régénérer les commentaires pour récupérer les nouveaux commentaires de chaque publication

En tant qu'agent, je souhaite sélectionner un commentaire pour l'afficher sur le meme offre dans le site en tanque un avis

En tant qu'agent, je souhaite voir le nombre de gouta restant pour la recuperations des nouveaux commentaires et le nombre pour les supprimer.

En tant qu'agent, je souhaite afficher certains commentaires sur le site comme avis client afin de rassurer les futurs clients.

En tant qu'agent, je souhaite consulter les réservations de voyage et afin de les refuser ou les accepter pour autoriser le paiement à l'utilisateur inscrit.

En tant qu'agent, je souhaite consulter les réservations de vols ou d'hôtels afin de les refuser ou les accepter pour autoriser le paiement à l'utilisateur inscrit.

En tant qu'agent, je souhaite consulter les demandes d'assurance soumises afin de les refuser ou les accepter pour autoriser le paiement à l'utilisateur inscrit.

En tant qu'agent, je souhaite consulter les demandes de visa (tourisme, étude, affaires) afin de les refuser ou les accepter pour autoriser le paiement à l'utilisateur inscrit.

En tant qu'agent, je souhaite valider ou refuser une réservation.

En tant qu'agent, je souhaite ajouter une ville pour pouvoir récupérer ses hotels

En tant qu'agent, je souhaite consulter les messages reçus par les utilisateurs inscrit et pouvoir répondre

## **Admin**

En tant qu'administrateur, je souhaite accéder à toutes les fonctionnalités et actions disponibles pour les agents, avec des droits étendus.

En tant qu'administrateur, je Souhaite Gérer mon compte personnel et modifier mes informations

En tant qu'administrateur, je Souhaite pouvoir Gérer les Agents(ajouter, modifier, supprimer)

En tant qu'administrateur, je veux consulter La liste des utilisateurs et filtrer par rôle et rechercher par nom et email

En tant qu'administrateur, je souhaite accéder aux factures reçus, les filtrer par type de service et les imprimer ou télécharger afin d'en assurer le suivi administratif.

En tant qu'administrateur, je souhaite modifier les clés API et passer le site en mode production afin de garantir son bon fonctionnement.

## **Utilisateur\_inscrit**

En tant qu'utilisateur inscrit, je souhaite créer un compte afin d'accéder aux services de l'agence.

En tant qu'utilisateur inscrit, je Souhaite Gérer mon compte personnel et modifier mes informations

En tant qu'utilisateur inscrit, Suivre mes réservations pour voir l'état de ma demande.

En tant qu'utilisateur inscrit, je souhaite consulter les publications de site (voyages/omras) voir les commentaires et nombre de likes récupérer depuis facebook et instagram pour la meme publication

En tant qu'utilisateur inscrit, je souhaite cliquer sur un commentaire et accéder directement vers la même publication sur Facebook ou Instagram

En tant qu'utilisateur inscrit, je souhaite réserver un voyage organisé ou une Omra en ligne et attendre que la réservation soit validée pour payer ensuite.

En tant qu'utilisateur inscrit, je souhaite réserver un vol ou un hôtel à partir d'un formulaire afin d'organiser mon voyage.

En tant qu'utilisateur inscrit, je souhaite voir l'emplacement de chaque hôtel à travers la carte Maps

En tant qu'utilisateur inscrit, je souhaite voir l'emplacement de l'agence à travers la carte Maps

En tant qu'utilisateur inscrit, je souhaite souscrire à une assurance en remplissant un formulaire avec pièces jointes afin de sécuriser mon voyage.

En tant qu'utilisateur inscrit, je souhaite faire une demande de visa en ligne en remplissant un formulaire avec pièces jointes afin de soumettre ma candidature

En tant qu'utilisateur inscrit, je souhaite contacter l'agence via e-mail, téléphone, messenger, formulaire de contact afin d'obtenir des informations ou de l'assistance.

En tant qu'utilisateur inscrit, je souhaite effectuer un paiement en DZD (carte CIB ou Dahabia) pour finaliser ma réservation de manière locale et sécurisée.

En tant qu'utilisateur inscrit, je souhaite payer en devise (carte Visa ou Mastercard) afin de réserver depuis l'étranger sans contrainte.

En tant que qu'utilisateur inscrit, je souhaite Recevoir la confirmation de réservation de l'admin/agent comme notification pour que le paiement soit autorisé ou non

## Client

En tant que client, je souhaite recevoir les factures de paiements et les télécharger

### 2.4.3 Backlog product

Le tableau 2.3 présente les différentes fonctionnalités identifiées pour le développement de l'application ZIGUADE TOUR. Chaque fonctionnalité est référencée afin de faciliter la gestion du projet lors des phases de planification et de développement.

Réf.	Fonctionnalité
Ref 1	Interface de gestion des utilisateurs pour l'administrateur et l'agent (ajout, modification, suppression, filtrage, recherche).
Ref 2	Interface de gestion des agents (ajouter, modifier, supprimer) accessible uniquement à l'administrateur.
Ref 3	Interface d'ajout d'un voyages organisés pour l'administrateur et l'agent.
Ref 4	Interface de gestion des voyages organisés (ajout, modification, suppression, publication) pour l'administrateur et l'agent.
Ref 5	Interface d'ajout d'une Omra pour l'administrateur et l'agent.
Ref 6	Interface de gestion des Omras (ajout, modification, suppression, publication) pour l'administrateur et l'agent.
Ref 7	Interface des publication des chaque offre(Omra et voyages organisés ) sur les réseaux sociaux (Facebook, Instagram,site) avec affichage du nombre restant de publications possibles au-dessus ( l'administrateur et l'agent.).
Ref 8	Interface de gestion des commentaires de chaque publication sur les réseaux sociaux(voir, supprimer, régénrer, sélectionner) avec avec affichage du nombre restant de récupération des nouveaux commentaires et le nombre restant pour les supprimer
Ref 9	Interface de détails de chaque voyage pour pour l'utilisateur inscrit.
Ref 10	Interface de la liste des voyages avec des commentaires et nombre de likes pour l'utilisateur inscrit.
Ref 11	Interface de réservation des voyages organisés pour l'utilisateur inscrit (formulaire + paiement en ligne si la demande est validée).
Ref 12	Interface de détails de chaque omra pour pour l'utilisateur inscrit.
Ref 13	Interface de réservation de Omra pour l'utilisateur inscrit (formulaire + paiement en ligne si la demande est validée).

Ref 14	Interface de la liste d'omra avec des commentaires et nombre de likes pour l'utilisateur inscrit.
Ref 15	Interface de recherche des hôtel
Ref 16	Interface des détails d'hôtel avec formulaire de réservation e la carte maps ( pour l'utilisateur inscrit.
Ref 17	Interface des détails une ville pour pouvoir récupérer ses hotels
Ref 18	Interface de recherche des vols
Ref 19	Interface de réservation de vol (formulaire, pièces jointes, paiement en ligne si la demande est validée) pour l'utilisateur inscrit.
Ref 20	Interface de gestion des réservations pour l'administrateur et l'agent (consulter, modifier, annuler, accepter/refuser).
Ref 21	Interface de suivi des paiements liés aux réservations et services ( état payé/non payé) (admin ).
Ref 22	Interface d'admin et d'agent pour le suivi et la validation ou le refus des reservations(voyage/omra/hotel/vol) .
Ref 23	Interface de gestion des demandes de visa pour l'utilisateur inscrit (remplissage, dépôt de documents, validation, paiement après validation de la demande).
Ref 24	Interface de gestion des assurances pour l'utilisateur inscrit (formulaire, pièces jointes, validation, paiement).
Ref 25	Interface pour le suivi et la validation ou le refus des demandes d'assurance (admin et agent).
Ref 27	Interface de traitement des demandes de visa : vérification des dossiers,accepter ou refuser une demande.
Ref 28	Interface de configuration des clés API (réservée à l'administrateur).
Ref 29	Interface d'inscription a la plateforme (pour visiteur).
Ref 30	Interface de connexion à la plateforme (pour tous les utilisateurs).
Ref 31	Interface "Mon espace" pour les utilisateurs inscrits et clients ( Modifier le profil, suivre leurs réservations, suivre l'état de leur demandes d'assurances et de visa ,Télécharger les factures de paiement , les réponses sur les messages).
Ref 32	Interface publique de consultation des avis clients mis en avant.
Ref 34	Interface d'accueil présentant les services, les offres populaires, et des accès rapides.
Ref 35	Interface "choix du paiement" présentant les différentes possibilités de paiement ( en agence, en DZD avec chargily ou en devise avec Stripe).
Ref 36	Interface de formulaire de contact pour utilisateur <sub>inscrit</sub>

TABLE 2.3 – Product Backlog de l'application ZIGUADE TOUR

#### 2.4.4 Sprint Backlog

Ce tableau 2.4 présente le Sprint Backlog, qui recense les différentes tâches dérivées du Backlog Produit mentionné précédemment.

Sprint	Items	En tant que	Je souhaite	Priorité
1	Authentification	Administrateur, utilisateur inscrit, Agent , client	-M'authentifier pour accéder à mon espace	1
	Inscription	Visiteur	Créer un compte	1
	Gestion des comptes Agents	Administrateur	- Ajouter, modifier, supprimer un compte d'Agent	1
	Gestion des utilisateurs inscrits	Administrateur	- Supprimer le compte d'utilisateur; <i>inscrit</i>	2
	Consultation et filtrage de la liste des utilisateurs	Administrateur	- Consulter les utilisateurs - Filtrer un utilisateur	2
	Gérer mon compte personnel	Administrateur, utilisateur inscrit, Agent , client	- Modifier mes informations personnelles	2
2	Gestion des voyages organisés	Admin / Agent	Creer, modifier, supprimer un voyage	1
	Gestion Omra	Admin / Agent	Creer, modifier, supprimer Omra	1
	Gestion des publications sur le site et les Réseaux sociaux	Admin / Agent	- Publier le meme offre voyage/Omra sur Site & Facebook & Instagram - Récupérer les publications de chaque offre et afficher le nombre restant pour les publications possible - Récupérer et filtrer la liste des voyages/omra par (publier/non publier),plateforme,mois de depart,année de depart	1

Gestion des commentaires	Admin / Agent	<ul style="list-style-type: none"> <li>- Récupérer les anciens commentaires de chaque publication depuis la Base de données</li> <li>- Récupérer les nouveaux commentaires de chaque publication depuis les réseaux sociaux</li> <li>- Supprimer un commentaire sur les réseaux sociaux</li> <li>- sélectionner des commentaires des réseaux et les faire comme des avis sur le site</li> <li>- l'affichage du nombre restant de récupération des nouveaux commentaires et le nombre restant pour les supprimer</li> <li>- accéder aux commentaires sur instagram ou facebook (la meme publication) depuis un clique sur l'avis qui se trouve dans le site</li> </ul>	1
Récupérer le nombre de réactions sur une publication	Admin / Agent	<ul style="list-style-type: none"> <li>- Récupérer le nombre de réactions sur une publication</li> <li>- l'affichage du nombre de réactions sur une publication du site</li> </ul>	1
Gestion des Réservations (voyages & Omra)	Utilisateur inscrit	<ul style="list-style-type: none"> <li>-Remplir un formulaire de réservation (voyages &amp; Omra)</li> <li>-Attendre la confirmation de la réservation pour procéder au paiement</li> </ul>	1

Gestion des réservations (vols & hôtels)	Utilisateur inscrit	-Remplir un formulaire de réservation(vols & hôtels) -Attendre la confirmation de la réservation pour procéder au paiement	1
Affichage des hôtels et vols (via API)	Utilisateur inscrit	-Récupérer les hotels et les vols a travers des recherches en temps réel -Ajouter le formulaire d'ajout d'une ville pour pouvoir récupérer ses hotels	1
Gestion des réservations	Admin / Agent	- Recevoir et consulter les réservation (vol/hotel/admin/agent) - Confirmer (autoriser le paiement) ou refuser une réservation pour (vol/hotel/admin/agent)	1
Filtrage des réservations	Admin / Agent	- Récupérer toutes les réservations - Filtrer par statut (payé / non payé), date, type	1
Panier utilisateur inscrit	utilisateur inscrit	- Recevoir les confirmation des reservations hotel/vol/omra/voyage - Recevoir les confirmation des demandes d'assurances/visas - Recevoir les réponses sur les messages envoyés - Confirmer (autoriser le paiement) ou refuser une réservation pour (vol/hotel/admin/agent)	1
Panier client	Client	recevoir les factures de paiements et les télécharger	1

	Configuration des API vols & hôtels	Admin	- Modifier les clés d'accès API - Passer en mode production	2
3	Messagerie (contact)	Utilisateur inscrit, Client	- Interface pour contacter l'agence via email, téléphone, WhatsApp, Messenger - Liens ou boutons cliquables vers les moyens de contact (email , messenger , instagram , numéro de téléphone)	1
	Messagerie (contact)	Agent , Admin	- répondre pour les messages reçu par utilisateur inscrit et client	1
	Assurance – Front réservation	Utilisateur inscrit	- Remplir un formulaire pour souscrire une assurance voyage (infos personnelles, dates, scan passeport) - Affichage dans une page dédiée	1
	Gestion des assurances	Admin / Agent	- Voir les demandes d'assurances reçues - Accepter (autoriser le paiement) ou refuser la demande d'assurance	1
	Visa – Front demande	Utilisateur inscrit	- Remplir un formulaire pour faire demande de visa (infos personnelles, dates, scan passeport) - Affichage dans une page dédiée	1

	Gestion des visas	Admin / Agent	- Consulter les demandes de visa reçus - Valider (Demander paiement) ou refuser les demandes de visa	1
--	-------------------	---------------	---	---

TABLE 2.4 – Sprint Backlog

## 2.5 Les Outils et Langages utilisés



**Postman** : Postman est une plateforme complète de développement d'API qui permet de concevoir, tester, documenter et collaborer sur des interfaces de programmation (APIs). Elle facilite chaque étape de création des API dans un même outil `postman.com`. [8]



**GitHub** : GitHub est une plateforme de développement basée sur le cloud permettant d'héberger et de partager du code source. Elle offre un contrôle de version Git et des outils de collaboration pour que des équipes de développeurs puissent travailler ensemble sur des projets logiciels. [9]



**Meta for Developers** Meta for Developers (plateforme de Meta /Facebook) fournit aux développeurs un ensemble d'API et d'outils pour intégrer les services de Meta (Facebook, Instagram, WhatsApp, etc.) dans leurs applications. Elle regroupe les bibliothèques (SDK) et interfaces nécessaires pour construire des fonctionnalités sociales et de messagerie. [10]



**phpMyAdmin** :phpMyAdmin est un outil open-source écrit en PHP qui offre une interface Web graphique pour administrer des bases de données MySQL ou MariaDB. Il permet de créer, modifier ou interroger des bases et tables (gestion des utilisateurs, exécution de requêtes SQL, etc.) sans passer par la ligne de commande. [11]



**Draw.io** : draw.io est un logiciel en ligne gratuit de dessin de diagrammes. On peut l'utiliser directement dans le navigateur pour réaliser des organigrammes, schémas de processus, diagrammes UML, diagrammes réseau, etc., et sauvegarder ses fichiers localement ou dans le cloud.



**Visual Studio Code** : Visual Studio Code est un éditeur de code open-source développé par Microsoft supportant un très grand nombre de langages grâce à des extensions. Il fournit aux développeurs à la fois un environnement de développement intégré avec des outils permettant de faire avancer les projets techniques. [12]



**React** : React est une bibliothèque JavaScript permettant de créer des interfaces utilisateur dynamiques. Elle facilite le développement d'applications web en utilisant des composants réutilisables. [13]



**Bootstrap** : Bootstrap est un framework développé par l'équipe du réseau social de Twitter. Disponible en open source, le framework fournit aux développeurs les outils nécessaires pour créer facilement des sites Web à l'aide des langages HTML, CSS et JavaScript. Ce dernier est conçu pour développer des sites Web avec un design réactif qui s'adapte à tout type d'écran. Il fournit des outils avec des styles existants pour la typographie, les boutons, les interfaces de navigation, etc. [14]



**Node.js** : Node.js est un environnement d'exécution JavaScript asynchrone et orienté événements, conçu pour créer des applications réseau évolutives. Il permet d'exécuter du code JavaScript côté serveur, ce qui est utile pour développer des sites web dynamiques et des services Web performants. [15]



**Express.js** : Le framework Express (également nommé Express.js) est un ensemble de modules Node permettant de créer facilement des applications web à partir de Node. Il est basé sur le modèle MVC (Model View Controller) qui permet de donner une architecture cohérente à une application web. Cette architecture MVC fait qu'Express est très populaire dans Node, car de plus en plus d'applications Node sont construites sur ces bases. [16]



**MySQL** : MySQL est un système de gestion de bases de données relationnelles (SGBDR) open source utilisé pour stocker et gérer des données. Sa fiabilité, ses performances, son évolutivité et sa facilité d'utilisation font de MySQL un choix plébiscité par les développeurs. [17]



**Sequelize** : Sequelize est un ORM (Object-Relational Mapping) moderne TypeScript et Node.js pour Oracle, Postgres, MySQL, MariaDB, SQLite, SQL Server, et bien plus encore. Il offre une prise en charge complète des transactions, des relations, des chargements anticipés et différés, de la réplication en lecture, et bien plus encore. [18]



**XAMPP** : XAMPP est une distribution Apache entièrement gratuite et facile à installer contenant MySQL, PHP et Perl. Le paquetage open source XAMPP a été mis au point pour être incroyablement facile à installer et à utiliser. [19]



**AJAX** : Ajax (Asynchronous JavaScript and XML) est une technique de développement web qui permet de récupérer du contenu depuis le serveur via des requêtes HTTP asynchrones. Elle met ensuite à jour les parties pertinentes de la page sans la recharger entièrement. [20]

## 2.6 Conclusion

Dans ce chapitre, nous avons défini les besoins pour le développement de l'application ZI-GUADE TOUR, en identifiant les acteurs et leurs rôles dans le projet. Cette étape nous a permis de mieux cerner les attentes des utilisateurs et de définir des exigences fonctionnelles adaptées à chaque type d'utilisateur. Le projet a été structuré selon la méthodologie Scrum, facilitant une gestion agile du développement. Nous avons détaillé les rôles des intervenants et les scénarios d'utilisation sous forme de user stories pour mieux comprendre les objectifs métiers. Le backlog produit a été créé en priorisant les fonctionnalités essentielles et en organisant le travail en sprints. Cette approche permet une gestion continue et une amélioration constante de l'application. Les bases solides posées ici nous permettent de passer à l'étape suivante avec des priorités bien définies. Dans le prochain chapitre, nous détaillerons l'application de la méthode Scrum pour la planification des tâches et livrables.

# Chapitre 3: Etude Sprint 1,2 et 3

## 3.1 Introduction

Après avoir identifié les besoins fonctionnels dans les chapitres précédents, nous entamons la phase de réalisation du projet à travers les trois premiers Sprints. Ce chapitre présente l'étude de chaque Sprint en détaillant les objectifs poursuivis, les modules développés ainsi que les aspects techniques abordés. Pour chaque partie, nous nous appuyons sur des descriptions textuelles et des diagrammes de modélisation tels que les cas d'utilisation, les diagrammes de classes ou de séquence.

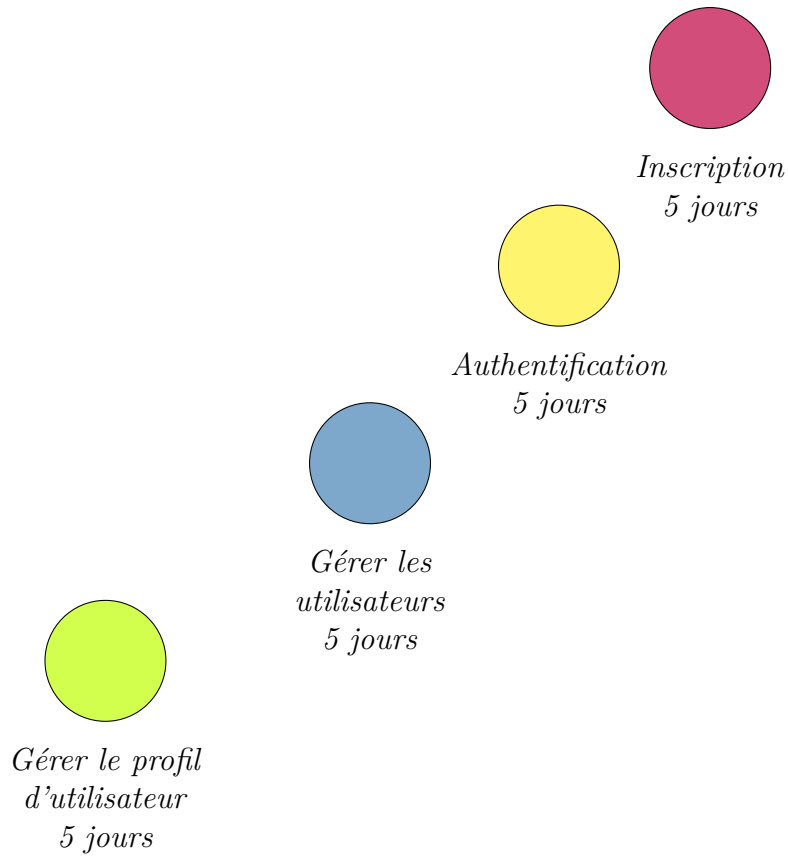
Cette étape concrétise le passage de la conception à la mise en œuvre, en assurant que chaque fonctionnalité corresponde aux besoins exprimés initialement.

## 3.2 Étude du 1er Sprint

On estime que ce premier sprint va durer **20 jours** et se compose de quatre items tels que :

- **Inscription**
- **Authentification**
- **Gérer les utilisateurs**
- **Gérer le profil d'utilisateur**

### 3.2.1 Décomposition du Sprint 1



### 3.2.2 Diagramme de cas d'utilisation global du Sprint 1

La figure suivante 3.1 montre le diagramme de cas d'utilisation globale de sprint 01.

Diagramme de cas D'utilisation 1Sprint

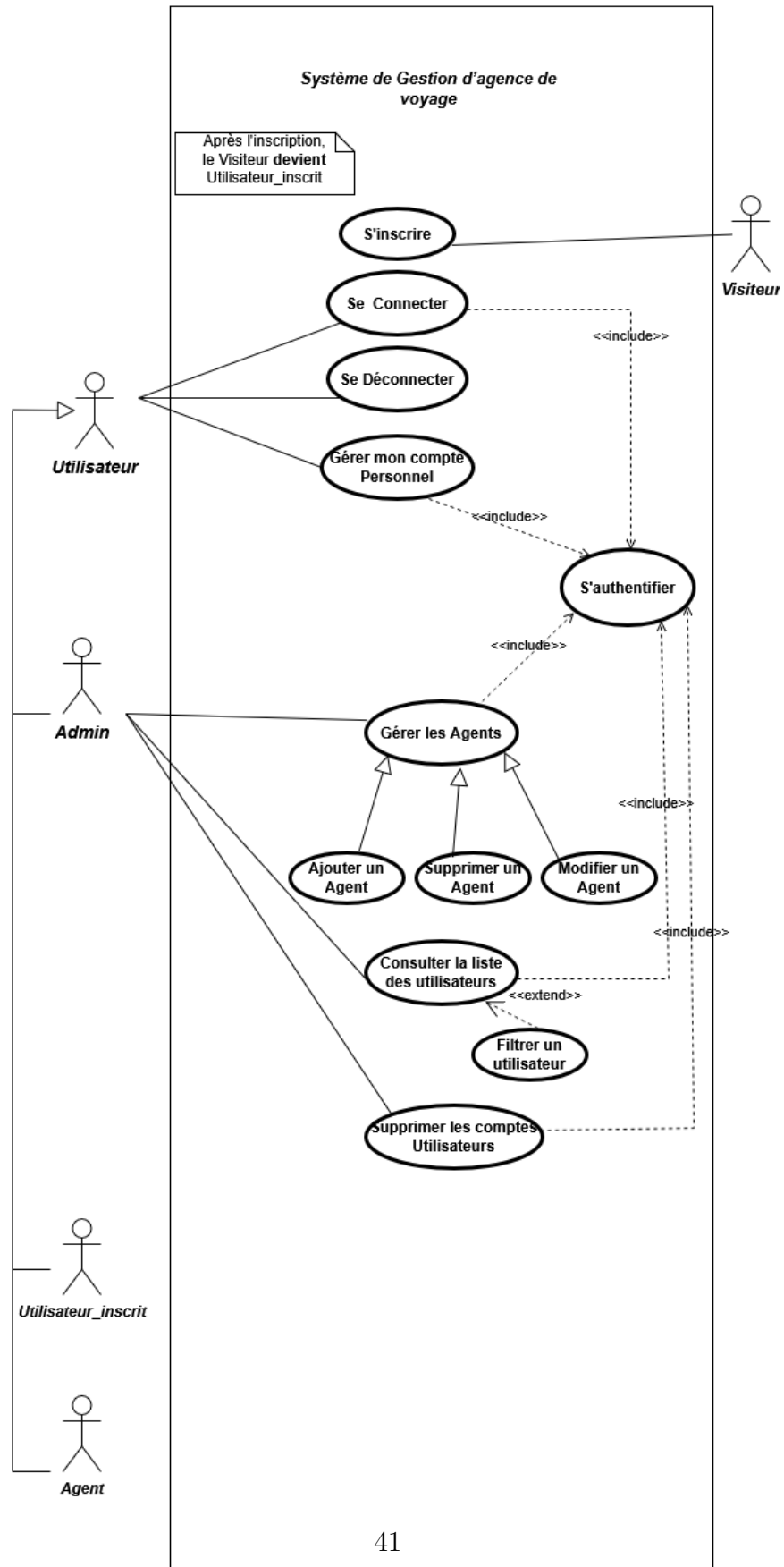


FIGURE 3.1 – diagramme de cas d'utilisation sprint1

### 3.2.3 Cas d'utilisation Inscription

#### 3.2.3.1 Description textuelle du cas d'utilisation « Inscription »

Le tableau 3.1 montre la description textuelle de cas d'utilisation « Inscription ».

<b>Cas d'utilisation</b>	S'inscrire
<b>But</b>	Permettre à un visiteur de créer un compte utilisateur en saisissant ses informations personnelles et en recevant un token JWT pour accéder à l'application.
<b>Acteur principal</b>	Visiteur
<b>Acteur secondaires</b>	/
<b>Préconditions</b>	Le visiteur n'a pas de compte existant avec l'adresse email utilisée.
<b>Scénario nominal</b>	<ol style="list-style-type: none"><li>1. Le visiteur demande la page d'inscription.</li><li>2. L'interface d'inscription affiche la page.</li><li>3. Le visiteur saisit ses informations puis clique sur "S'inscrire".</li><li>4. L'interface vérifie la validité des champs.</li><li>5. Si les champs sont valides, l'interface envoie les informations au service d'inscription.</li><li>6. Le service vérifie que l'email n'existe pas dans la base.</li><li>7. Si l'email est inexistant, le service enregistre le nouvel utilisateur.</li><li>8. Le service génère un token. Le service retourne un message de succès et le token.</li><li>9. L'interface affiche la confirmation d'inscription.</li></ol>
<b>Enchaînements alternatifs</b>	<ul style="list-style-type: none"><li>— <b>Si les champs sont invalides</b>, l'interface demande au visiteur de resaisir les informations.</li><li>— <b>Si l'email existe déjà</b>, le service renvoie un message d'erreur, affiché ensuite à l'utilisateur.</li></ul>
<b>Postconditions</b>	Un nouvel utilisateur est enregistré avec succès et reçoit un token JWT pour accéder à l'application.

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

#### 3.2.3.2 Diagramme de séquence du cas d'utilisation « Inscription »

La figure 3.2 représente le diagramme de séquence de cas d'utilisation « Inscription ».

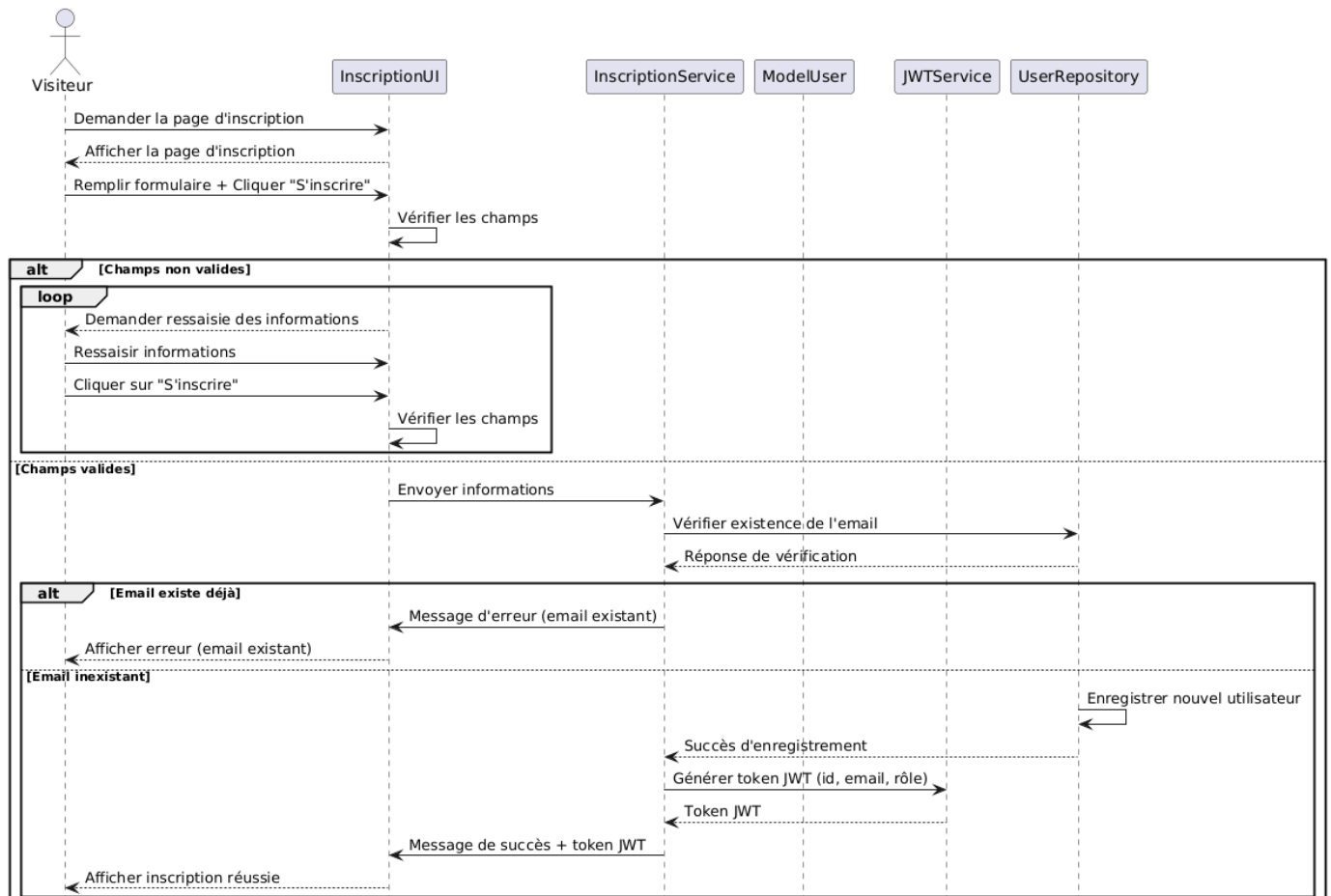


FIGURE 3.2 – Diagramme de Séquence Inscription

### 3.2.4 Cas d'utilisation Authentification

#### 3.2.4.1 Description textuelle du cas d'utilisation « Authentification »

Le tableau 3.2 montre la description textuelle du cas d'utilisation « Authentification ».

<b>Cas d'utilisation</b>	S'authentifier
<b>But</b>	Permettre à un utilisateur inscrit de se connecter à l'application en saisissant ses identifiants.
<b>Acteur principal</b>	Utilisateur inscrit
<b>Acteur secondaires</b>	/
<b>Préconditions</b>	L'utilisateur possède déjà un compte valide (adresse email et mot de passe enregistrés).

<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande la page de connexion.</li> <li>2. L'interface affiche la page de connexion.</li> <li>3. L'utilisateur saisit ses identifiants et clique sur "Se connecter".</li> <li>4. L'interface vérifie que les champs sont valides.</li> <li>5. Si les champs sont valides, les identifiants sont transmis au contrôleur d'authentification.</li> <li>6. Le contrôleur interroge le service utilisateur pour valider les informations.</li> <li>7. Si elles sont correctes, le service retourne les informations de l'utilisateur et un token JWT.</li> <li>8. L'interface affiche un message de succès et accorde l'accès à l'utilisateur.</li> </ol>
<b>Enchaînements alternatifs</b>	<ul style="list-style-type: none"> <li>— <b>Si les champs sont invalides ou manquants</b>, l'interface demande la ressaisie ou affiche une erreur.</li> <li>— <b>Si les identifiants sont incorrects</b>, un message d'erreur est affiché à l'utilisateur.</li> </ul>
<b>Postconditions</b>	L'utilisateur est connecté et dispose d'un token JWT valide pour accéder aux fonctionnalités de l'application.

TABLE 3.2 – Description textuelle du cas d'utilisation « Authentification »

### 3.2.4.2 Diagramme de séquence du cas d'utilisation « Authentification »

La figure 3.3 le diagramme de séquence de cas d'utilisation « Authentification ».

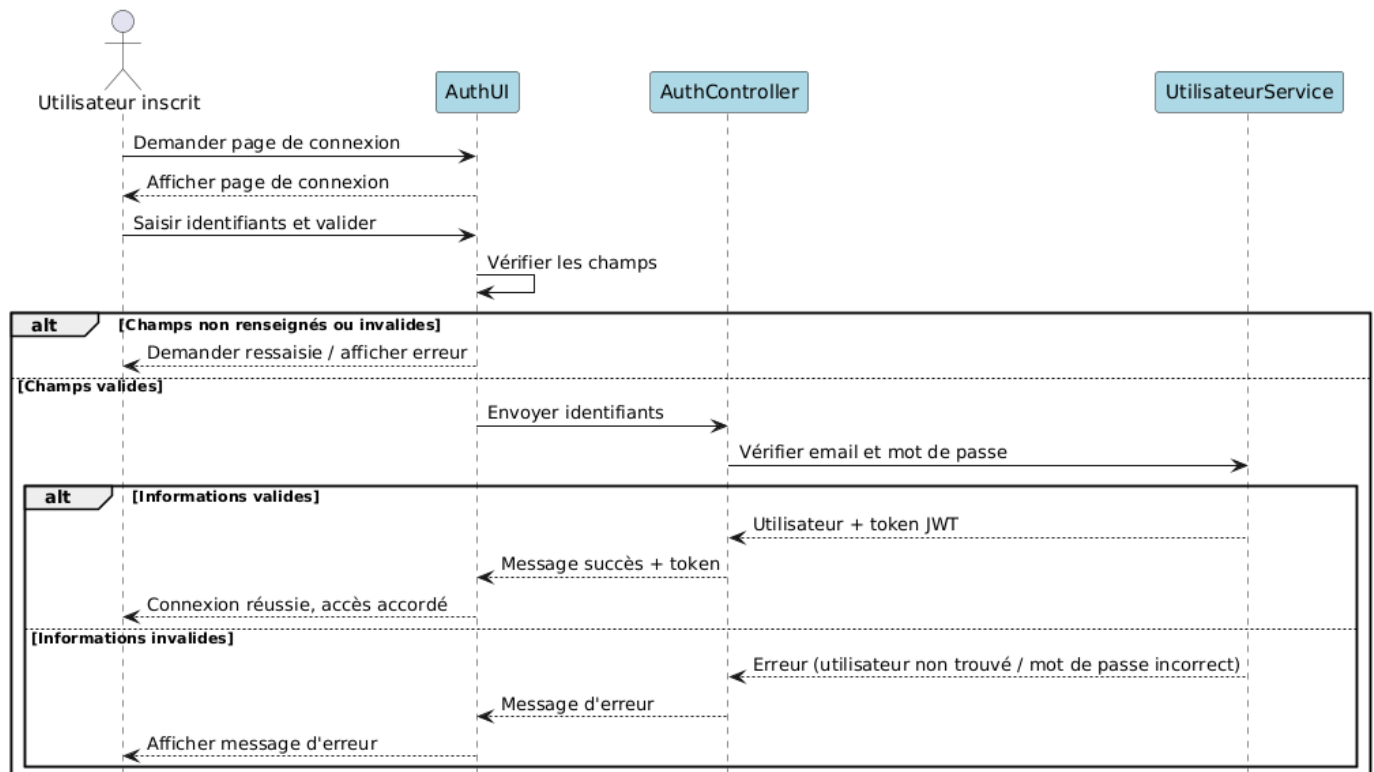


FIGURE 3.3 – Diagramme de Séquence Authentification

### 3.2.5 Cas d'utilisation Gérer mon compte

#### 3.2.5.1 Description textuelle du cas d'utilisation « Gérer mon compte »

Le tableau 3.3 montre la description textuelle du cas d'utilisation « Gérer mon compte ».

<b>Cas d'utilisation</b>	Gérer mon compte
<b>But</b>	Mise à jour des informations personnelles de l'utilisateur.
<b>Acteur principal</b>	Utilisateur inscrit
<b>Acteur secondaires</b>	/
<b>Préconditions</b>	L'utilisateur est connecté et possède un compte valide.

<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur accède à la page « Mon compte ».</li> <li>2. L'interface affiche les informations actuelles de l'utilisateur.</li> <li>3. L'utilisateur modifie les champs souhaités.</li> <li>4. L'utilisateur clique sur « Enregistrer ».</li> <li>5. Le formulaire est envoyé au contrôleur via une requête POST.</li> <li>6. Le contrôleur vérifie la validité des champs.</li> <li>7. Si les champs sont valides, les informations sont mises à jour en base de données.</li> <li>8. Le contrôleur renvoie une confirmation de succès.</li> <li>9. L'interface affiche un message de confirmation.</li> </ol>
<b>Enchaînements alternatifs</b>	<ul style="list-style-type: none"> <li>— <b>Si un ou plusieurs champs sont invalides</b>, le contrôleur renvoie un message d'erreur.</li> <li>— <b>L'interface</b> affiche le message d'erreur et invite l'utilisateur à corriger les informations.</li> </ul>
<b>Postconditions</b>	Les informations de l'utilisateur sont mises à jour en base de données.

TABLE 3.3 – Description textuelle du cas d'utilisation « Gérer mon compte »

### 3.2.5.2 Diagramme de séquence du cas d'utilisation « Gérer mon compte »

La figure 3.4 montre le diagramme de séquence de cas d'utilisation « Gérer mon compte ».

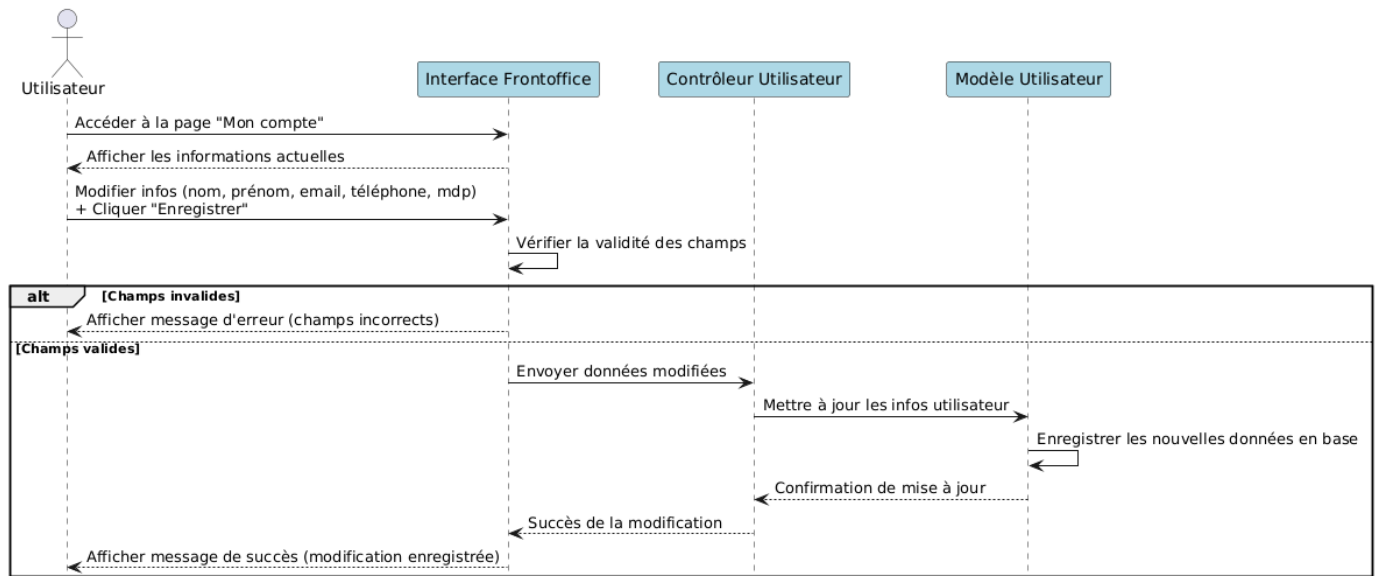


FIGURE 3.4 – Diagramme de Séquence Gérer mon compte

### 3.2.6 Cas d'utilisation Ajouter un agent

#### 3.2.6.1 Description textuelle du cas d'utilisation « Ajouter un agent »

Le tableau 3.4 montre la description textuelle du cas d'utilisation « Ajouter un agent ».

<b>Cas d'utilisation</b>	Ajouter un agent
<b>But</b>	Permettre à l'administrateur d'enregistrer un nouvel agent avec ses informations personnelles et professionnelles.
<b>Acteur principal</b>	Administrateur
<b>Acteur secondaires</b>	/
<b>Préconditions</b>	<ul style="list-style-type: none"> <li>— L'administrateur est authentifié.</li> <li>— L'email de l'agent n'existe pas déjà dans la base de données.</li> </ul>

<p><b>Scénario nominal</b></p>	<ol style="list-style-type: none"> <li>1. L'administrateur accède au formulaire d'ajout d'agent.</li> <li>2. Le système affiche un formulaire vide avec les champs requis (nom, prénom, email, téléphone, matricule...).</li> <li>3. L'administrateur remplit le formulaire et le soumet.</li> <li>4. Le contrôleur vérifie que l'email est unique.</li> <li>5. Si l'email est disponible, le système crée d'abord un utilisateur avec le rôle "agent".</li> <li>6. Ensuite, il crée les informations spécifiques à l'agent dans la table Agent.</li> <li>7. Le système retourne une réponse 201 et affiche un message de succès avec les données créées.</li> </ol>
<p><b>Enchaînements alternatifs</b></p>	<ul style="list-style-type: none"> <li>— <b>Si l'email est déjà utilisé :</b> <ul style="list-style-type: none"> <li>— Le contrôleur retourne une erreur 400.</li> <li>— Le système affiche le message d'erreur : « Email déjà utilisé ».</li> </ul> </li> </ul>
<p><b>Postconditions</b></p>	<ul style="list-style-type: none"> <li>— Un nouveau compte agent est créé avec un enregistrement dans la table Utilisateur et la table Agent.</li> <li>— En cas d'échec, aucune donnée n'est enregistrée.</li> </ul>

TABLE 3.4 – Description textuelle du cas d'utilisation « Ajouter un agent »

### 3.2.6.2 Diagramme de séquence du cas d'utilisation « Ajouter un agent »

La figure 3.5 montre le diagramme de séquence de cas d'utilisation « Ajouter un agent ».

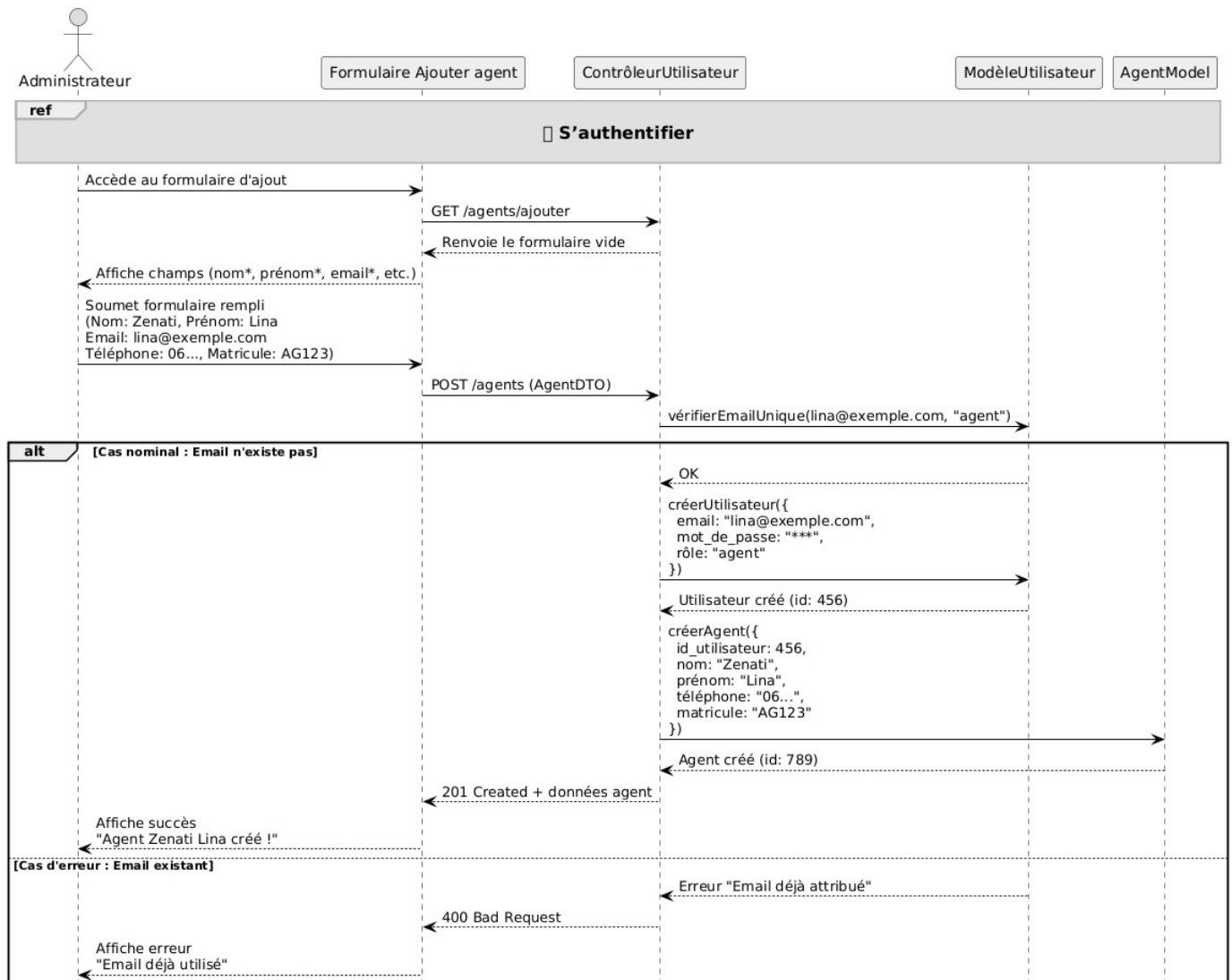


FIGURE 3.5 – Diagramme de Séquence Ajouter un agent

### 3.2.7 Cas d'utilisation Modifier un agent

#### 3.2.7.1 Description textuelle du cas d'utilisation « Modifier un agent »

Le tableau 3.5 montre la description textuelle du cas d'utilisation « Modifier un agent ».

<b>Cas d'utilisation</b>	Modifier un agent
<b>But</b>	Permettre à l'administrateur de mettre à jour les informations d'un agent (à la fois dans les tables Utilisateur et Agent).
<b>Acteur principal</b>	Administrateur
<b>Acteur secondaires</b>	/

<b>Préconditions</b>	<ul style="list-style-type: none"> <li>— L'administrateur est authentifié.</li> <li>— L'agent (utilisateur avec rôle = "agent") existe dans la base de données.</li> </ul>
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'administrateur consulte la liste des utilisateurs.</li> <li>2. Il sélectionne un utilisateur avec le rôle "agent".</li> <li>3. Il remplit le formulaire de modification.</li> <li>4. Le système récupère l'utilisateur par identifiant.</li> <li>5. Le système met à jour les champs dans la table Utilisateur (nom, prénom, email, téléphone).</li> <li>6. Le système récupère et met à jour les champs spécifiques dans la table Agent (matricule, date d'embauche).</li> <li>7. Un message de succès est affiché à l'administrateur.</li> </ol>
<b>Enchaînements alternatifs</b>	<ul style="list-style-type: none"> <li>— <b>Si l'utilisateur n'existe pas</b> : le système affiche le message « Utilisateur introuvable ».</li> <li>— <b>Si le rôle est différent de "agent"</b> : le système affiche le message « Modification refusée : non agent ».</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>— Les informations dans les deux tables Utilisateur et Agent sont mises à jour avec succès.</li> <li>— En cas d'erreur, aucune modification n'est enregistrée.</li> </ul>

TABLE 3.5 – Description textuelle du cas d'utilisation « Modifier un agent »

### 3.2.7.2 Diagramme de séquence du cas d'utilisation « Modifier un agent »

La figure 3.6 montre le diagramme de séquence de cas d'utilisation « Modifier un agent ».

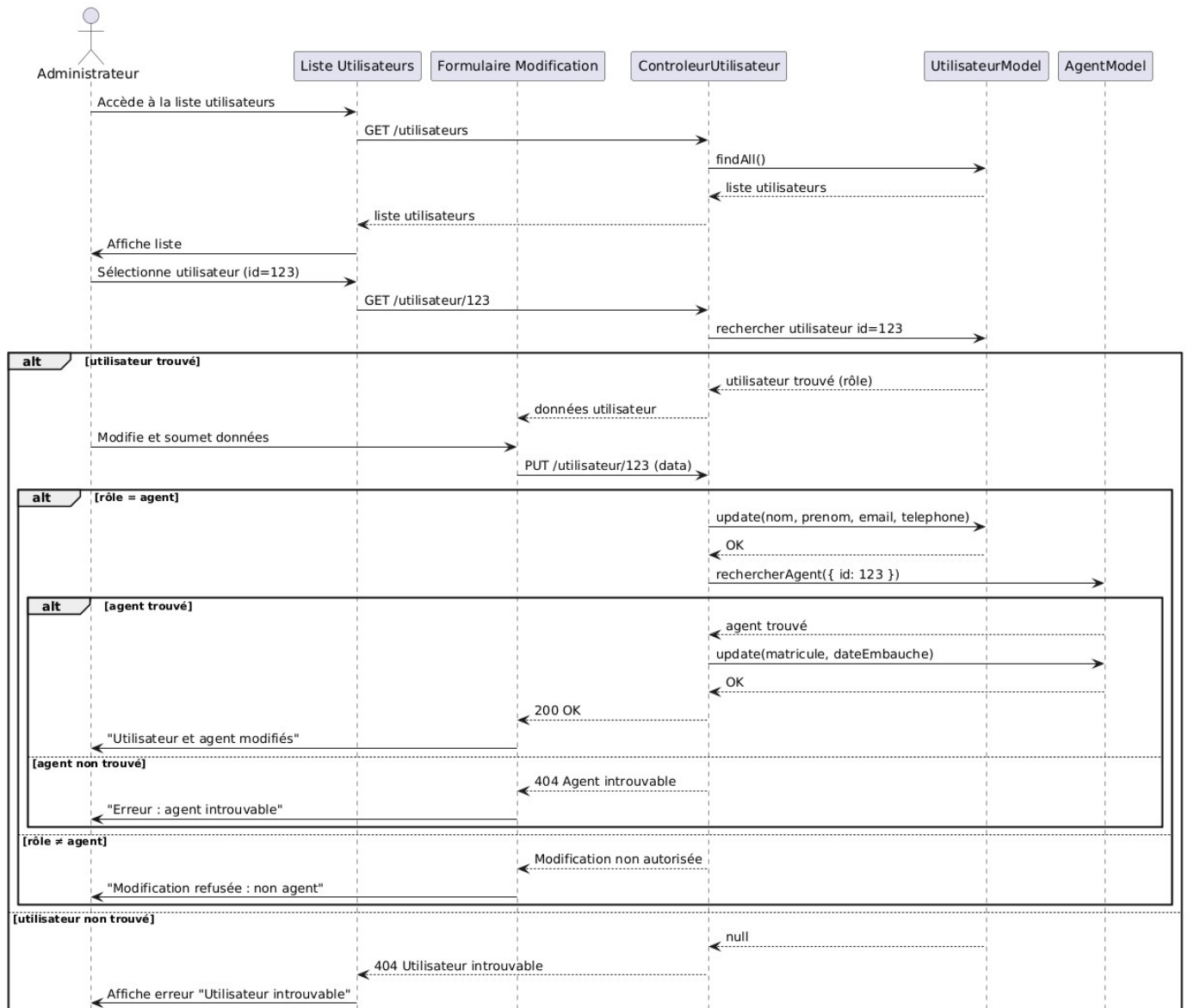


FIGURE 3.6 – Diagramme de Séquence Modifier un agent

### 3.2.8 Cas d'utilisation Supprimer un utilisateur inscrit

#### 3.2.8.1 Description textuelle du cas d'utilisation « Supprimer un utilisateur inscrit »

Le tableau 3.6 montre la description textuelle du cas d'utilisation « Supprimer un utilisateur inscrit ».

Cas d'utilisation	Supprimer un utilisateur inscrit
-------------------	----------------------------------

<b>But</b>	Permettre à l'administrateur de supprimer un utilisateur de type "utilisateur inscrit" s'il n'a pas été connecté depuis plus de 2 ans et qu'il n'a pas de réservation.
<b>Acteur principal</b>	Administrateur
<b>Acteur secondaires</b>	/
<b>Préconditions</b>	<ul style="list-style-type: none"> <li>— L'administrateur est authentifié.</li> <li>— L'utilisateur à supprimer existe dans la base de données.</li> </ul>
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'administrateur accède à la liste des utilisateurs.</li> <li>2. Il clique sur « Supprimer » pour un utilisateur (exemple : id 456).</li> <li>3. Le système vérifie que l'utilisateur existe et que son rôle est « utilisateur inscrit ».</li> <li>4. Le système vérifie que l'utilisateur est inactif depuis plus de 2 ans.</li> <li>5. Le système vérifie qu'il n'a aucune réservation.</li> <li>6. Le système supprime les données dans la table UtilisateurInscrit.</li> <li>7. Le système supprime les données dans la table Utilisateur.</li> <li>8. Un message de succès est affiché à l'administrateur.</li> </ol>
<b>Enchaînements alternatifs</b>	<ul style="list-style-type: none"> <li>— <b>Si l'utilisateur n'existe pas</b> : un message « Utilisateur introuvable » est affiché.</li> <li>— <b>Si l'utilisateur a un rôle différent</b> : message « Rôle non supprimable ».</li> <li>— <b>Si l'utilisateur est actif depuis moins de 2 ans</b> : message « Suppression interdite avant 2 ans ».</li> <li>— <b>Si l'utilisateur a des réservations</b> : message « Utilisateur non supprimé : réservations existantes ».</li> </ul>

<b>Postconditions</b>	<ul style="list-style-type: none"> <li>— L'utilisateur est supprimé seulement si toutes les conditions sont remplies.</li> <li>— Sinon, aucune suppression n'est effectuée.</li> </ul>
-----------------------	--

TABLE 3.6 – Description textuelle du cas d'utilisation « Supprimer un utilisateur inscrit »

### 3.2.8.2 Diagramme de séquence du cas d'utilisation « Supprimer un utilisateur inscrit »

La figure 3.7 montre le diagramme de séquence de cas d'utilisation « Supprimer un utilisateur inscrit ».

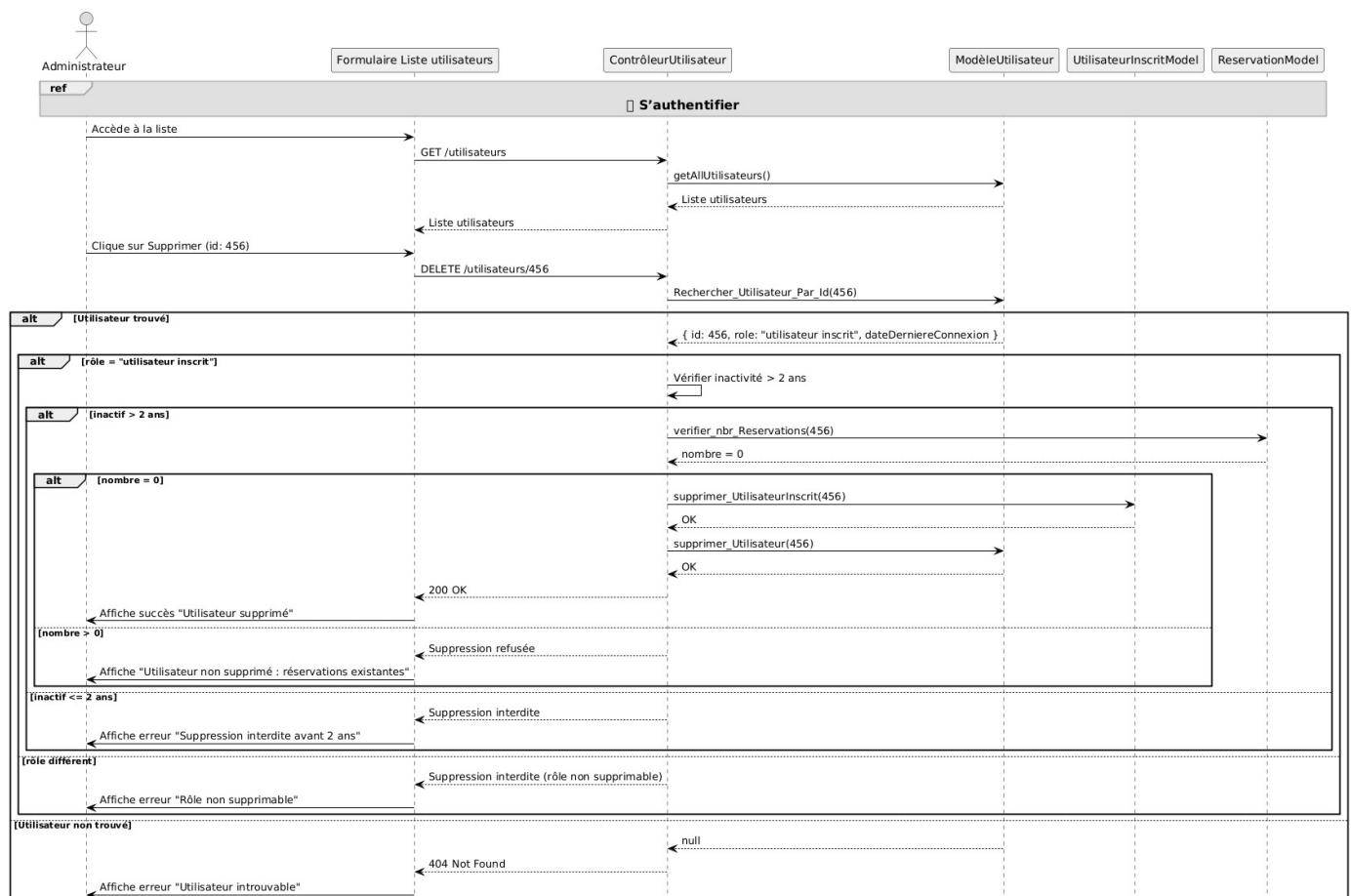


FIGURE 3.7 – Diagramme de Séquence Supprimer un utilisateur inscrit

### 3.2.9 Diagramme de Classe premier sprint

La figure 3.8 montre le diagramme de de Classe premier sprint.

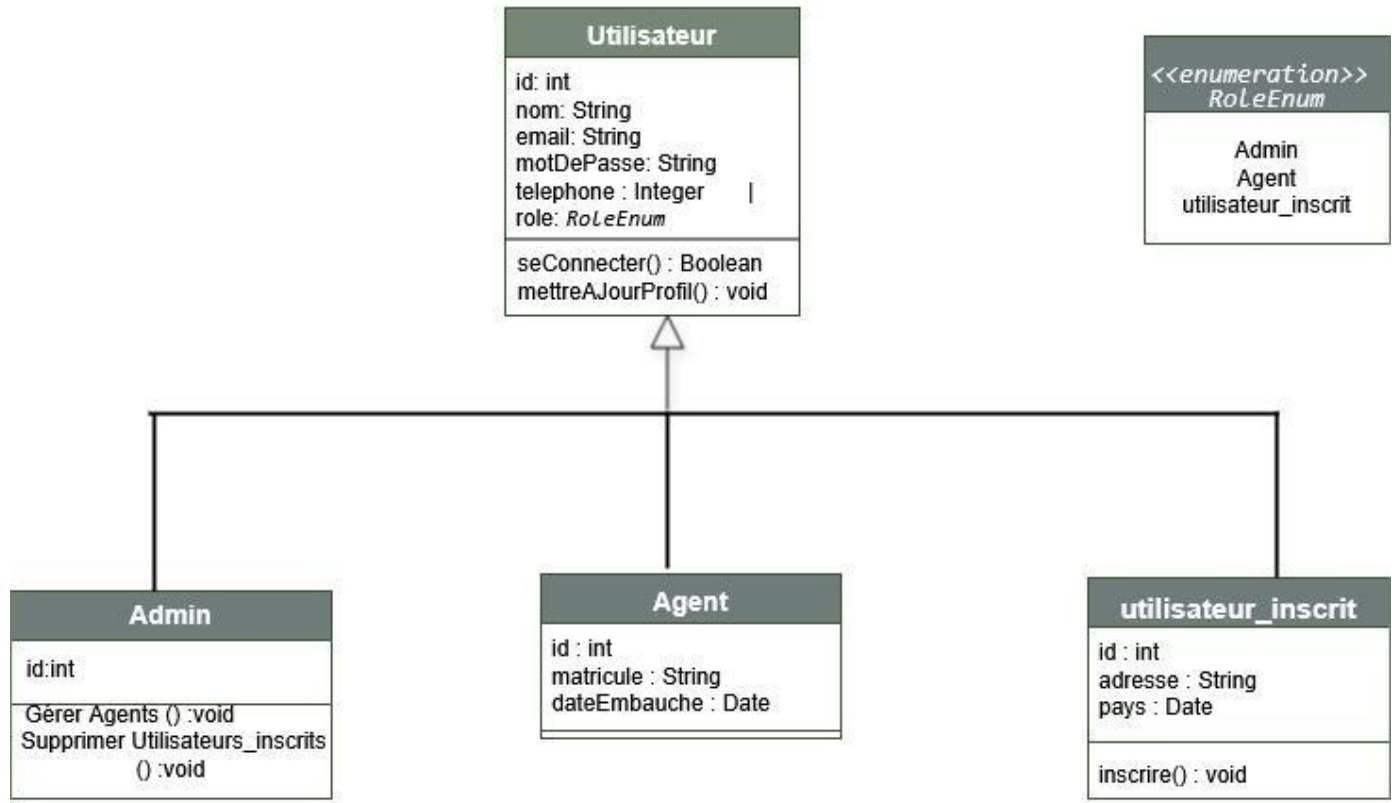


FIGURE 3.8 – Diagramme de Classe Sprint 1

## Modèle relationnel du 1<sup>er</sup> sprint

Voici le modèle relationnel correspondant au diagramme de classes du premier sprint :

- **Utilisateur**(id, nom, email, mot\_de\_passe, téléphone, rôle)
- **Utilisateur\_inscrit**(#id, adresse, pays)  
id est une clé étrangère vers **Utilisateur**(id)
- **Agent**(#id, matricule, date\_embauche)  
id est à la fois clé primaire et clé étrangère vers **Utilisateur**(id)
- **Admin**(#id)  
id est à la fois clé primaire et clé étrangère vers **Utilisateur**(id)

### 3.2.10 Interfaces du premier sprint

#### 3.2.10.1 Interface De la page d'accueil

Après la connexion, l'utilisateur inscrit accède à la page d'accueil de l'application. .



FIGURE 3.9 – Interface De la page d'accueil

### 3.2.10.2 Interface d'inscription

Le visiteur peut s'inscrire en fournissant ses informations personnelles (nom, prénom, email, mot de passe, etc.) afin de créer un compte et accéder aux services proposés par l'application.

FIGURE 3.10 – Interface d'inscription utilisateur

### 3.2.10.3 Interface d'authentification

Après authentification, l'utilisateur inscrit accède à la page d'accueil du front office, tandis que l'administrateur et l'agent sont redirigés vers le back office.

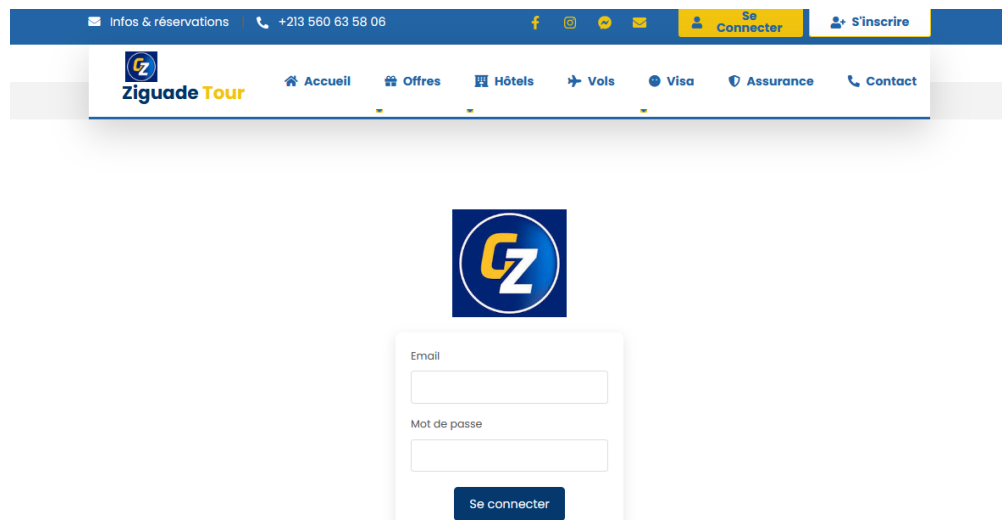


FIGURE 3.11 – Interface d’authentification utilisateur

#### 3.2.10.4 Interface Gérer mon espace

L'utilisateur inscrit peut accéder à son espace personnel afin de consulter et modifier ses informations (nom, prénom, email, mot de passe, etc.), suivre ses réservations, assurances et demandes de visa, recevoir les réponses des agents à ses messages, ainsi que consulter ses factures.

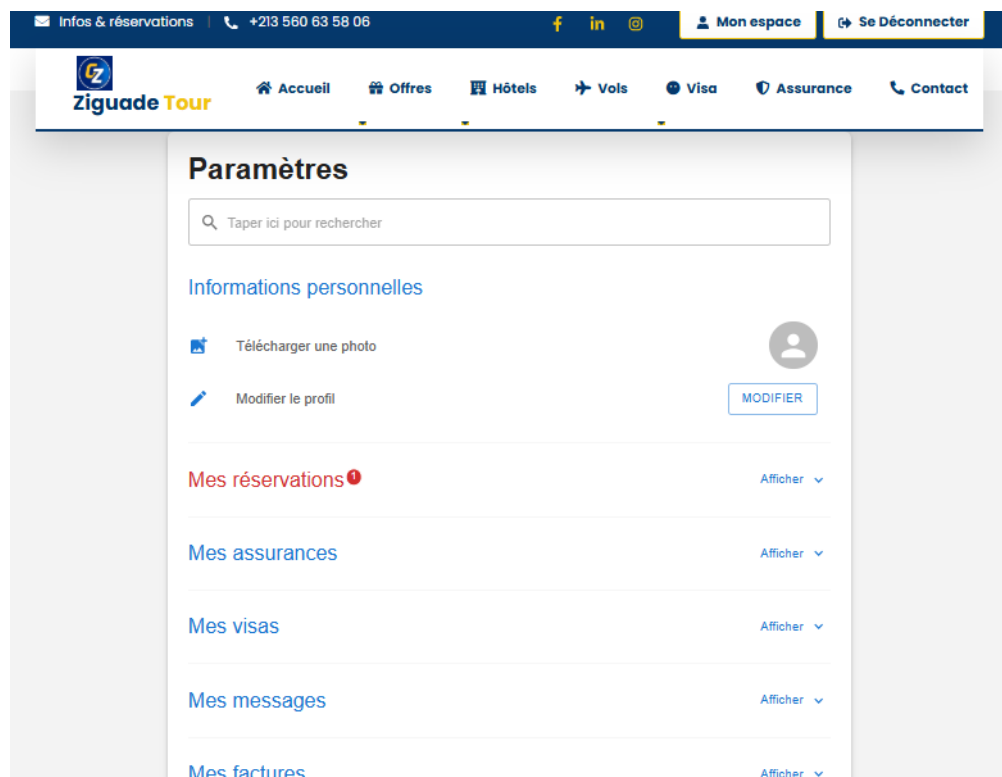


FIGURE 3.12 – Interface de gestion de l’espace utilisateur

### 3.2.10.5 Interface Ajouter un agent

L'administrateur peut ajouter un nouvel agent depuis le back-office en remplissant un formulaire contenant les informations nécessaires.

Explorez le monde avec Ziguade Tour

Retour au site +221 33 123 45 67 contact@ziguadetour.com

### Ajouter un Agent

\* Nom

\* Prénom

\* Email

\* Téléphone

\* Matricule

\* Date d'embauche

Sélectionner u...

Ajouter

© 2025 Ziguade Tour. Tous droits réservés.  
Créé avec pour les passionnés de voyage.

FIGURE 3.13 – Interface d'ajout d'un agent

### 3.2.10.6 Interface consulter la liste des utilisateurs inscrits

L'administrateur peut consulter la liste des utilisateurs inscrits .

Explorez le monde avec Ziguade Tour

Retour au site +221 33 123 45 67 contact@ziguadetour.com

Rechercher un utilisateur...

Tous les rôles

5

ID	Nom	Email	Rôle	Actions
44	hjn jbn	zianirimas@gmail.com	Utilisateur_inscrit	Modifier Supprimer
43	zenati lina	zianiab@gmail.com	Utilisateur_inscrit	Modifier Supprimer
42	ziani djamila	zianidjamil@gmail.com	Utilisateur_inscrit	Modifier Supprimer
41	zenati assile	zenatiiline2002@gmail.com	agent	Modifier Supprimer
40	zenati lina	znot@gmail.com	agent	Modifier Supprimer

Précédent Page 1 / 7 Suivant

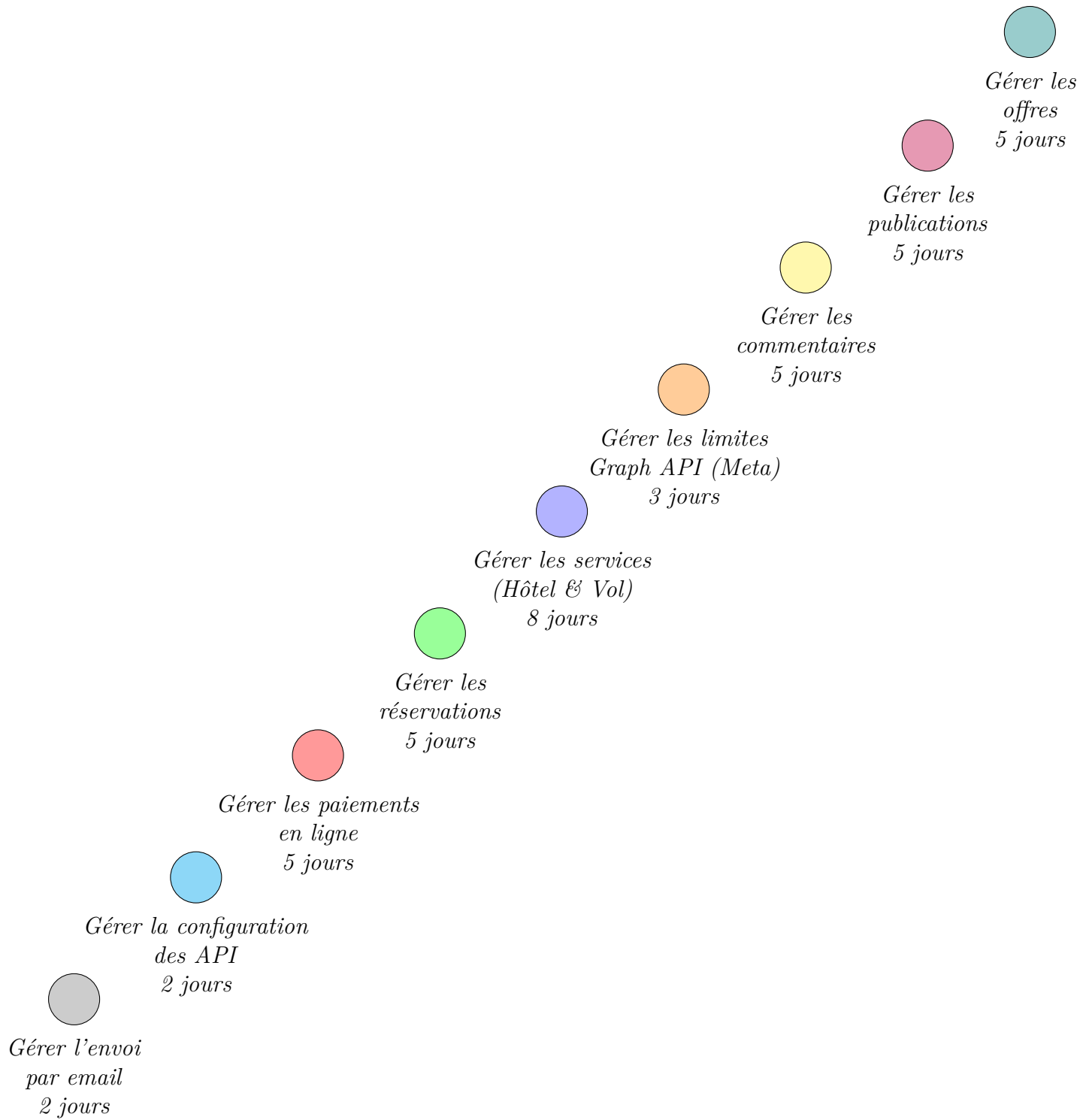
© 2025 Ziguade Tour. Tous droits réservés.  
Créé avec pour les passionnés de voyage.

FIGURE 3.14 – Interface des utilisateurs inscrits

## 3.3 Etude de 2ème Sprint

On estime que ce deuxième sprint va durer **35 jours** et se compose de neuf items tels que :

- Gérer les offres (Voyage Omra)
- Gérer les publications (site, facebook,instagram)
- Gérer les commentaires
- Gérer les réservations
- Gérer les limites de graph api (meta)
- Gérer les services (HotelVol)
- Gérer les paiements en ligne( chargily , stripe )
- Gérer la configuration des api
- Gérer l'envoi par email



### 3.3.1 Diagramme de cas d'utilisation Globale Sprint 2

La figure suivante 3.15 représente le diagramme de séquence de cas d'utilisation Globale Sprint 2.

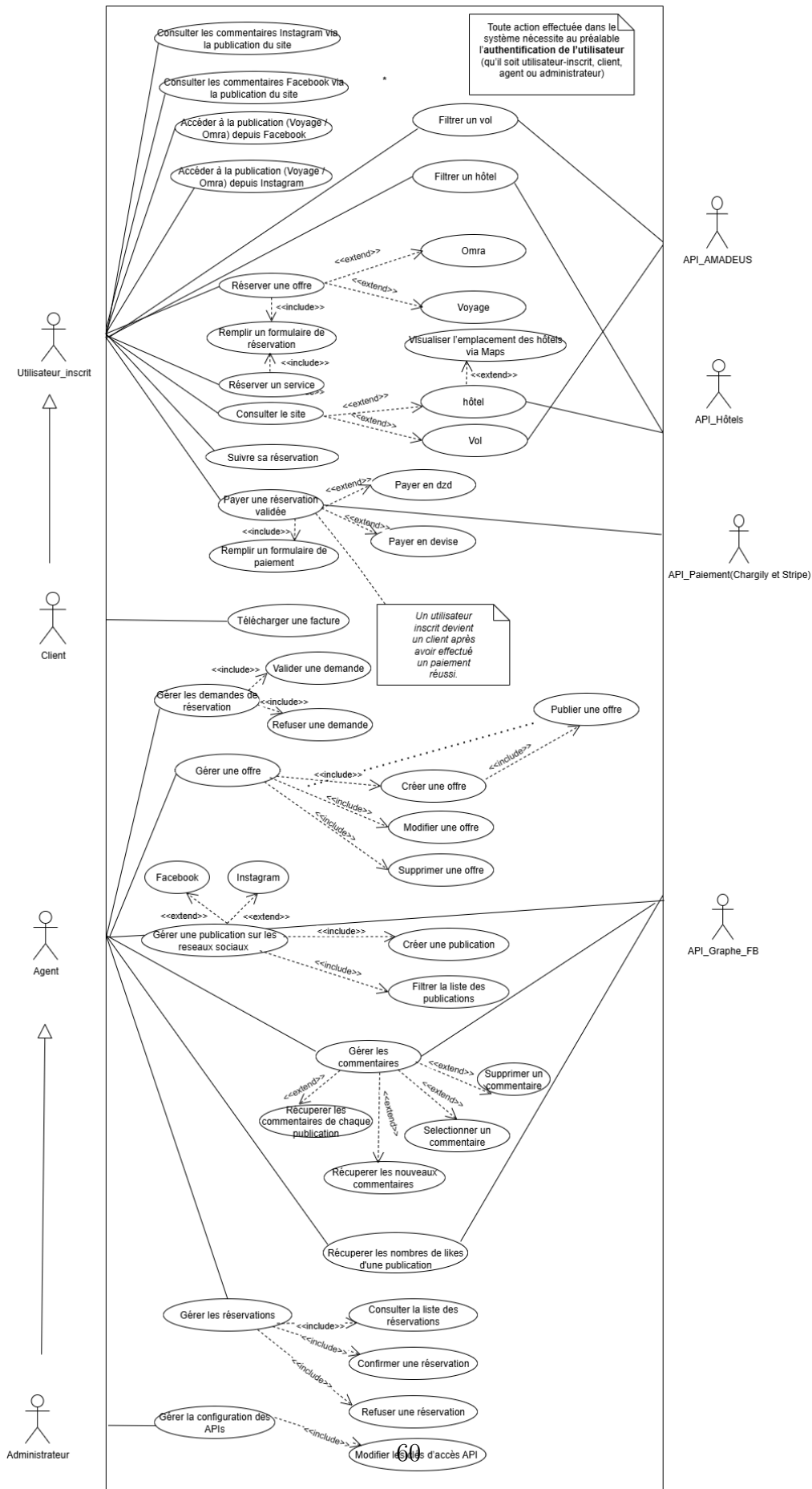


FIGURE 3.15 – Diagramme de cas d'utilisation sprint2

### 3.3.2 Cas d'utilisation Créer un voyage

#### 3.3.2.1 Description textuelle du cas d'utilisation « Créer un voyage »

Le tableau 3.7 montre la description textuelle du cas d'utilisation « Créer un voyage ».

<b>Cas d'utilisation</b>	Créer un voyage
<b>But</b>	Permettre à un administrateur ou un agent d'ajouter un nouveau voyage à l'aide d'un formulaire, avec enregistrement des données en base.
<b>Acteur principal</b>	Administrateur / Agent
<b>Acteurs secondaires</b>	/
<b>Préconditions</b>	<ul style="list-style-type: none"><li>— L'administrateur ou l'agent est authentifié.</li><li>— Le formulaire d'ajout est accessible.</li><li>— La connexion à la base de données est opérationnelle.</li></ul>
<b>Scénario nominal</b>	<ol style="list-style-type: none"><li>1. L'agent ou l'administrateur accède au formulaire.</li><li>2. Le système affiche les champs à remplir.</li><li>3. L'agent ou l'administrateur soumet les données.</li><li>4. Le contrôleur crée le voyage.</li><li>5. Le système retourne une réponse 201 et affiche un message de succès.</li></ol>
<b>Enchaînements alternatifs</b>	<ul style="list-style-type: none"><li>— <b>Si un champ est manquant ou invalide :</b><ul style="list-style-type: none"><li>— Sequelize retourne une erreur.</li><li>— Le contrôleur envoie un code 400.</li><li>— L'interface affiche un message d'échec.</li></ul></li></ul>
<b>Postconditions</b>	<ul style="list-style-type: none"><li>— En cas de succès : le voyage est enregistré.</li><li>— En cas d'échec : aucun voyage n'est créé et un message d'erreur est affiché.</li></ul>

TABLE 3.7 – Description textuelle du cas d'utilisation « Créer un voyage »

#### 3.3.2.2 Diagramme de séquence du cas d'utilisation « Créer un Voyage »

La figure montre le diagramme de séquence de cas d'utilisation « Créer un Voyage ».

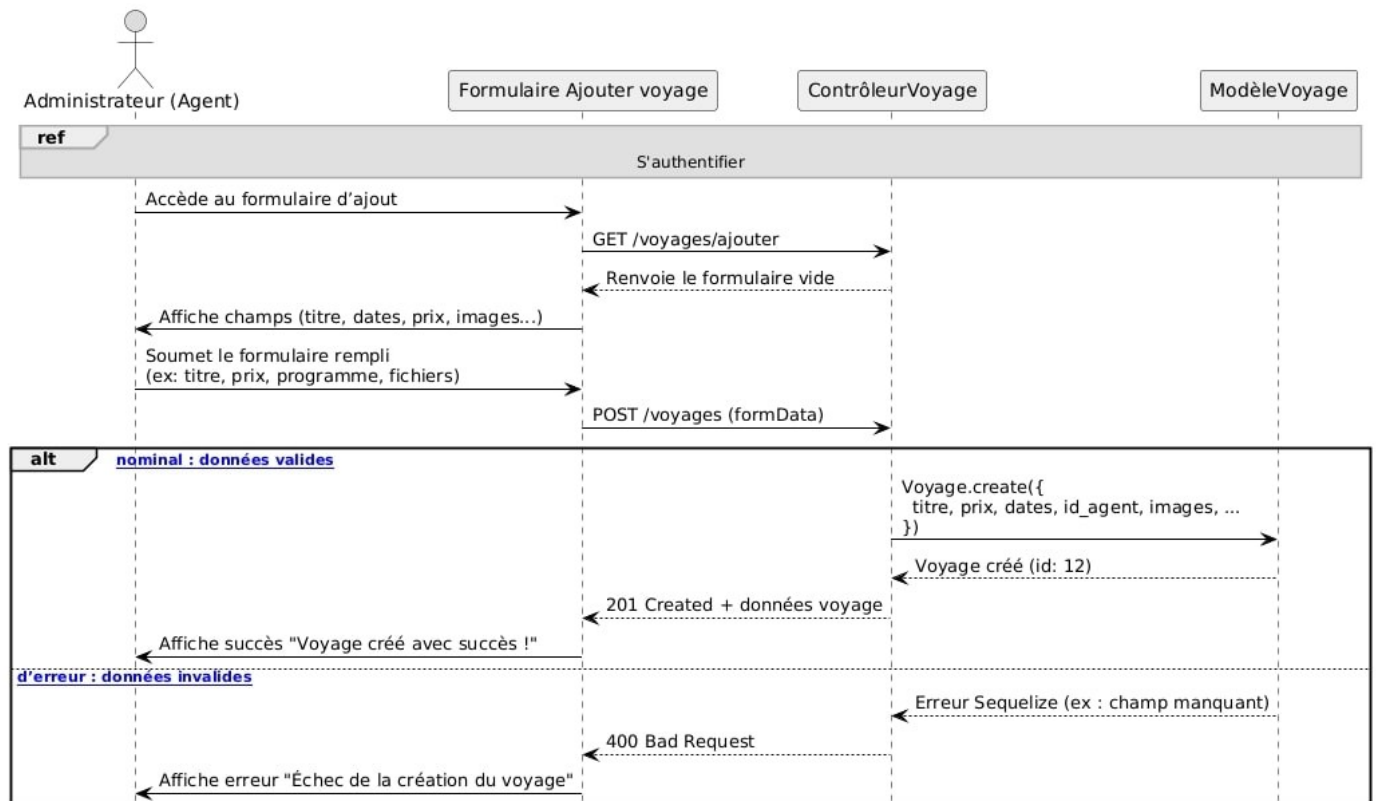


FIGURE 3.16 – Diagramme de Séquence Créer un Voyage

### 3.3.3 Cas d'utilisation Modifier un voyage

#### 3.3.3.1 Description textuelle du cas d'utilisation « Modifier un voyage »

Le tableau 3.8 montre la description textuelle du cas d'utilisation « Modifier un voyage ».

<b>Cas d'utilisation</b>	Modifier un voyage
<b>But</b>	Permettre à un administrateur ou un agent de mettre à jour les informations d'un voyage, à condition qu'il ne soit pas déjà publié et que sa date de départ ne soit pas dépassée.
<b>Acteur principal</b>	Administrateur / Agent
<b>Acteur secondaires</b>	/
<b>Préconditions</b>	<ul style="list-style-type: none"> <li>— L'administrateur ou l'agent est authentifié.</li> <li>— L'ID du voyage à modifier est connu.</li> <li>— Le formulaire de modification est accessible.</li> </ul>

<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'agent ou l'administrateur soumet la modification via PUT /voyages/{id} avec les nouvelles données.</li> <li>2. Le contrôleur recherche le voyage correspondant.</li> <li>3. Si le voyage est trouvé, le système vérifie son état.</li> <li>4. Si le voyage est modifiable (non publié et date de départ non dépassée), la mise à jour est effectuée.</li> <li>5. Le système retourne une réponse 200 OK avec confirmation.</li> </ol>
<b>Enchaînements alternatifs</b>	<ul style="list-style-type: none"> <li>— <b>Voyage non trouvé</b> : réponse 500 avec message « Voyage non trouvé ».</li> <li>— <b>Voyage publié ou date dépassée</b> : réponse 500 avec message « Modification interdite ».</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>— En cas de succès : le voyage est mis à jour en base.</li> <li>— En cas d'échec : aucune modification n'est effectuée, un message d'erreur est retourné.</li> </ul>

TABLE 3.8 – Description textuelle du cas d'utilisation « Modifier un voyage »

### 3.3.3.2 Diagramme de séquence du cas d'utilisation « Modifier un voyage »

La figure 3.17 montre le diagramme de séquence de cas d'utilisation « Modifier un voyage ».

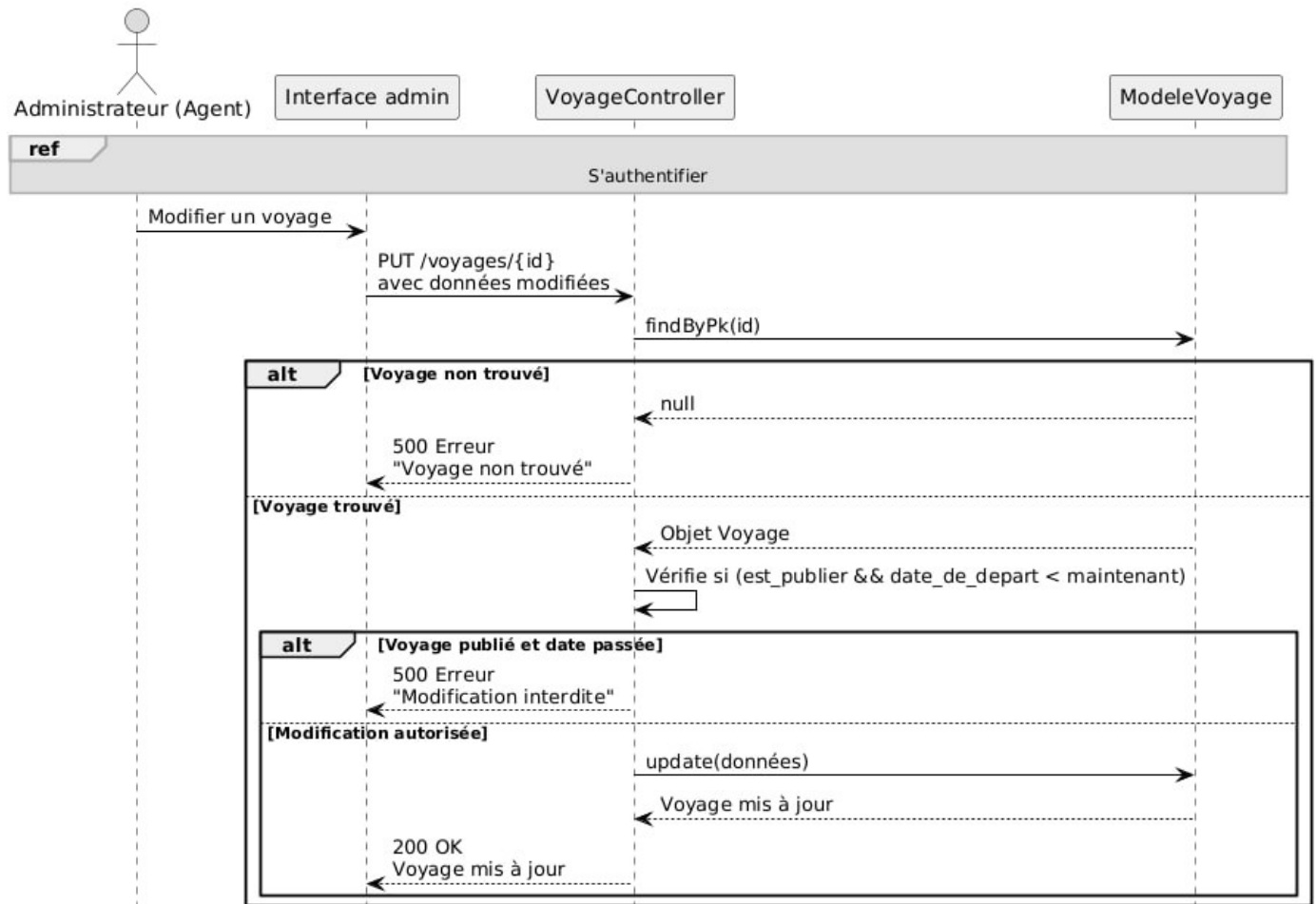


FIGURE 3.17 – Diagramme de Séquence Modifier un voyage

### 3.3.4 Cas d'utilisation Publier un Voyage sur Facebook

#### 3.3.4.1 Description textuelle du cas d'utilisation « Publier un Voyage sur Facebook »

Le tableau 3.9 montre la description textuelle de cas d'utilisation « Publier un Voyage sur Facebook ».

<b>Cas d'utilisation</b>	Publier un voyage sur Facebook
<b>But</b>	Permettre à un administrateur ou agent de publier un voyage sur Facebook en respectant certaines conditions (date valide, pas déjà publié, quota d'appels non dépassé).
<b>Acteur principal</b>	Administrateur / Agent
<b>Acteur secondaires</b>	/

<b>Préconditions</b>	<ol style="list-style-type: none"> <li>1. L'administrateur ou l'agent est authentifié.</li> <li>2. Le voyage à publier existe.</li> <li>3. La date de départ du voyage n'est pas dépassée.</li> <li>4. Le voyage n'a pas encore été publié sur Facebook.</li> </ol>
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'agent clique sur « Publier sur Facebook ».</li> <li>2. Le système récupère les détails du voyage.</li> <li>3. Le système vérifie que le voyage n'a pas déjà été publié.</li> <li>4. Le système vérifie que le quota d'appels API Facebook n'est pas dépassé.</li> <li>5. Si tout est OK, le système crée la publication (texte + images).</li> <li>6. Le système sauvegarde le post_id.</li> <li>7. Le système met à jour le statut du voyage comme publié.</li> <li>8. Le système retourne un message de succès (200 OK).</li> </ol>
<b>Enchaînements alternatifs</b>	<ul style="list-style-type: none"> <li>— Date dépassée : affichage de l'erreur « Impossible de publier, date dépassée ».</li> <li>— Voyage déjà publié : affichage du message « Voyage déjà publié ».</li> <li>— Quota d'appels dépassé : réponse 429 Too Many Requests.</li> <li>— Erreur API Facebook : réponse 400 Bad Request avec message d'erreur.</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>— En cas de succès : le voyage est publié sur Facebook, la publication est enregistrée, et le statut du voyage est mis à jour.</li> <li>— En cas d'échec : un message d'erreur explicite est retourné.</li> </ul>

TABLE 3.9 – Description textuelle du cas d'utilisation « Publier un Voyage sur Facebook »

### 3.3.4.2 Diagramme de séquence du cas d'utilisation « Publier un Voyage sur facebook »

La figure 3.18 montre le diagramme de séquence de cas d'utilisation « Publier un Voyage sur facebook ».

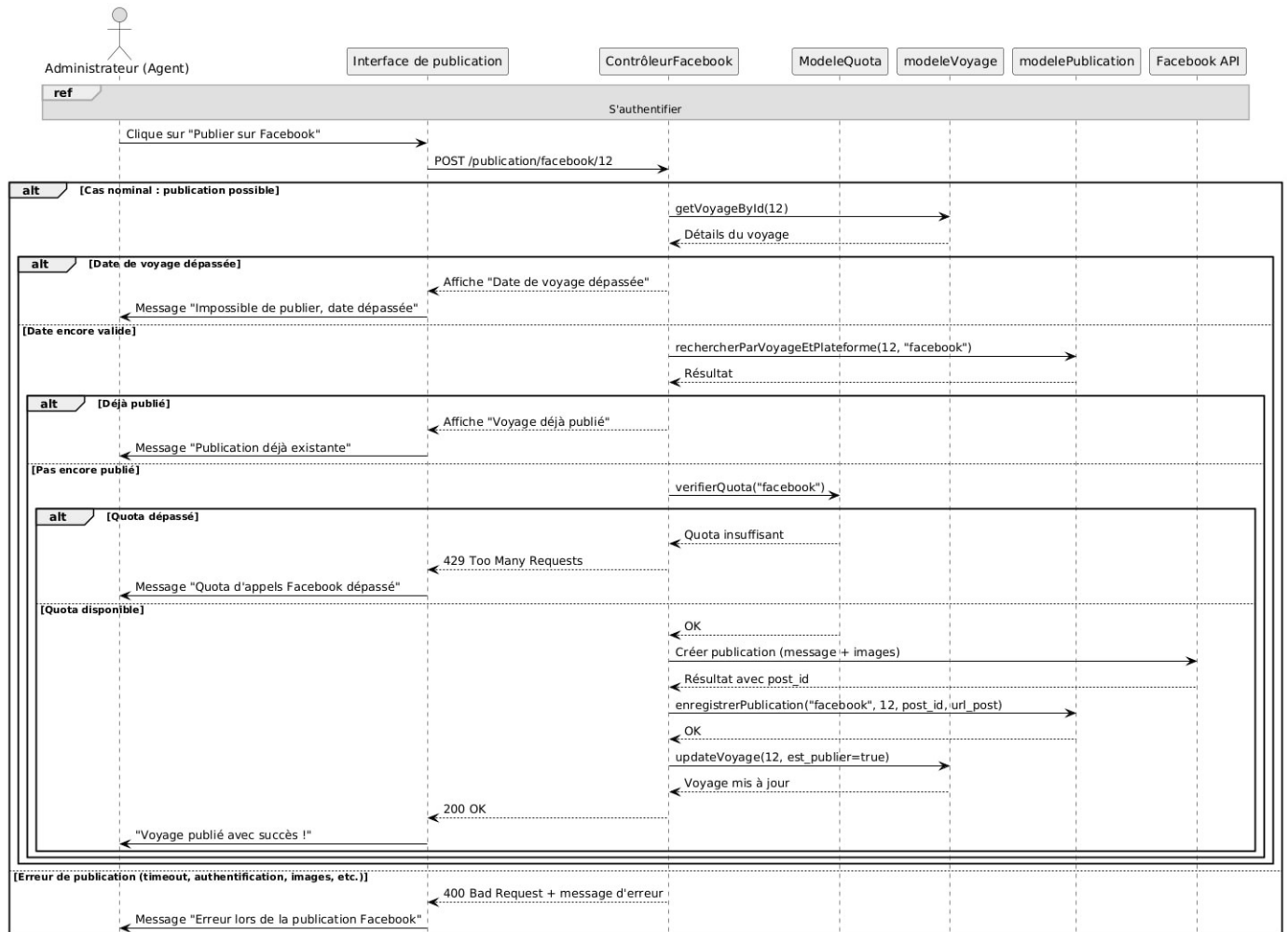


FIGURE 3.18 – Diagramme de Séquence Publier un Voyage sur facebook

### 3.3.5 Cas d'utilisation Récupérer les nouveaux Commentaires

#### 3.3.5.1 Description textuelle de cas d'utilisation « Récupérer les nouveaux Commentaires »

Le tableau 3.10 montre la description textuelle du cas d'utilisation « Récupérer les nouveaux Commentaires ».

<b>Cas d'utilisation</b>	Récupérer les nouveaux commentaires
<b>But</b>	Permettre à un administrateur ou à un agent de récupérer et d'afficher les nouveaux commentaires liés à une publication Facebook ou Instagram.
<b>Acteur principal</b>	Administrateur / Agent
<b>Acteurs secondaires</b>	/
<b>Préconditions</b>	<ul style="list-style-type: none"> <li>— L'utilisateur est authentifié.</li> <li>— La publication est connue.</li> <li>— Le quota de l'API n'est pas dépassé.</li> </ul>
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur clique sur « Afficher les commentaires ».</li> <li>2. Le système vérifie si le quota API est disponible.</li> <li>3. Si oui, l'API Facebook/Instagram est appelée.</li> <li>4. Les nouveaux commentaires sont enregistrés avec l'attribut <code>est_nouveau = true</code>.</li> <li>5. Le système envoie les nouveaux commentaires à l'interface administrateur.</li> <li>6. L'utilisateur consulte les commentaires et déclenche leur marquage comme lus (<code>est_nouveau = false</code>).</li> </ol>
<b>Enchaînements alternatifs</b>	<p><b>[Quota insuffisant] :</b></p> <ul style="list-style-type: none"> <li>— L'API renvoie une erreur 429 « Quota API dépassé ».</li> <li>— Le système affiche un message d'erreur.</li> </ul> <p><b>[Aucun nouveau commentaire] :</b></p> <ul style="list-style-type: none"> <li>— Le système récupère les anciens commentaires (<code>est_nouveau = false</code>).</li> <li>— Ils sont affichés à l'utilisateur.</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>— Les nouveaux commentaires sont affichés et marqués comme consultés.</li> <li>— Aucun appel à l'API n'est effectué si le quota est insuffisant.</li> </ul>

### 3.3.5.2 Diagramme de séquence de Cas d'utilisation « Récupérer les nouveaux Commentaires »

La figure 3.19 montre le diagramme de séquence de cas d'utilisation « Récupérer les nouveaux Commentaires ».

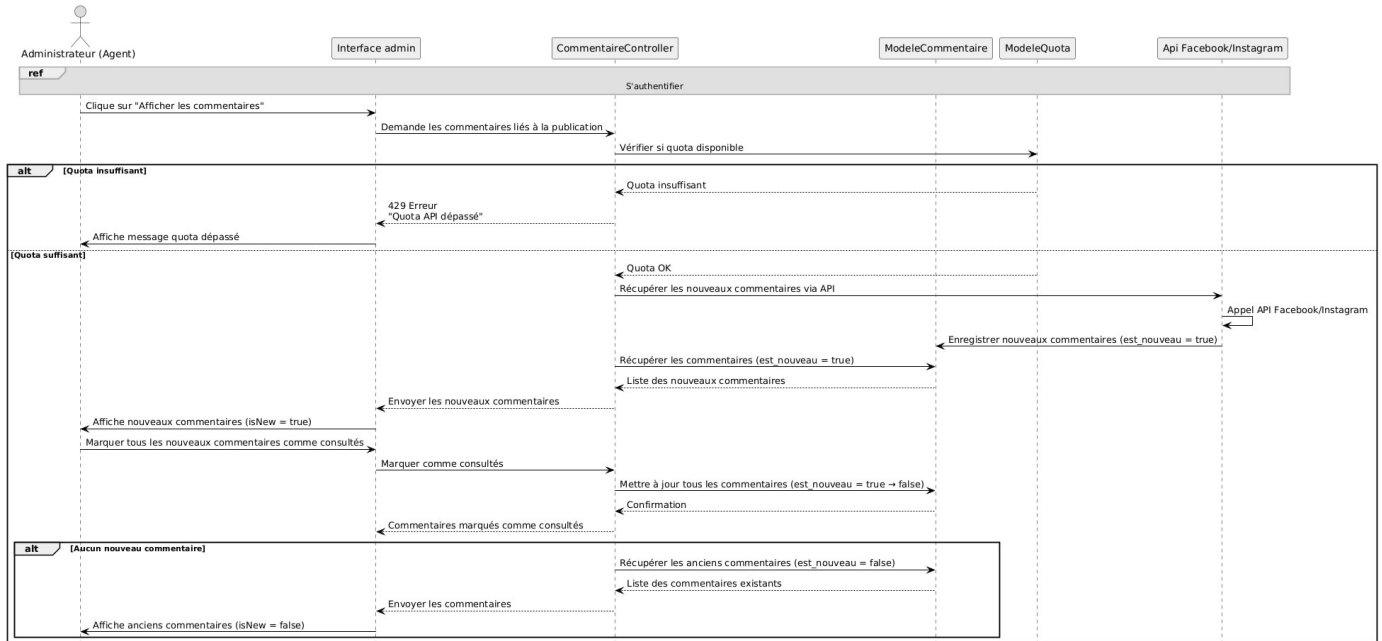


FIGURE 3.19 – Diagramme de séquence - Récupérer les nouveaux commentaires

### 3.3.6 Cas d'utilisation Sélectionner des Commentaires

#### 3.3.6.1 Description textuelle de cas d'utilisation Sélectionner des Commentaires

Le tableau 3.11 montre la description textuelle de cas d'utilisation « Sélectionner des Commentaires ».

<b>Cas d'utilisation</b>	Sélectionner / Désélectionner un commentaire
<b>But</b>	Permettre à un administrateur ou un agent de marquer un commentaire comme sélectionné ou désélectionné afin de le traiter ou de le mettre en avant.
<b>Acteur principal</b>	Administrateur / Agent
<b>Acteurs secondaires</b>	/
<b>Préconditions</b>	<ul style="list-style-type: none"> <li>— L'utilisateur est authentifié</li> <li>— Le commentaire est déjà existant dans la base de données</li> </ul>
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'administrateur coche ou décoche un commentaire via l'interface</li> <li>2. L'interface envoie une requête POST vers /commentaire/selection avec les données nécessaires</li> <li>3. Le contrôleur valide les champs et identifie la plateforme (Facebook ou Instagram)</li> <li>4. Le commentaire est recherché dans la base de données via son identifiant</li> <li>5. Si trouvé, son champ est_selectionne est mis à jour</li> <li>6. Un message de confirmation est retourné « Commentaire sélectionné / désélectionné »</li> </ol>
<b>Enchaînement alternatif</b>	<p><b>[Champs manquants ou invalides] :</b></p> <ul style="list-style-type: none"> <li>— Retour d'erreur 400 Bad Request avec message « Champs manquants ou invalides »</li> </ul> <p><b>[Plateforme inconnue] :</b></p> <ul style="list-style-type: none"> <li>— Retour d'erreur 500 avec message d'exception « Plateforme inconnue »</li> </ul> <p><b>[Commentaire non trouvé] :</b></p> <ul style="list-style-type: none"> <li>— Retour d'erreur 500 avec message d'exception « Commentaire non trouvé »</li> </ul>
<b>Postcondition</b>	Le commentaire est correctement marqué comme sélectionné ou désélectionné, ou une erreur est retournée si une condition n'est pas remplie.

TABLE 3.11 – Description textuelle du cas d'utilisation Sélectionner des Commentaires

### 3.3.6.2 Diagramme de séquence de Cas d'utilisation « Sélectionner des Commentaires »

La figure 3.20 montre le diagramme de séquence de cas d'utilisation « Sélectionner des Commentaires ».

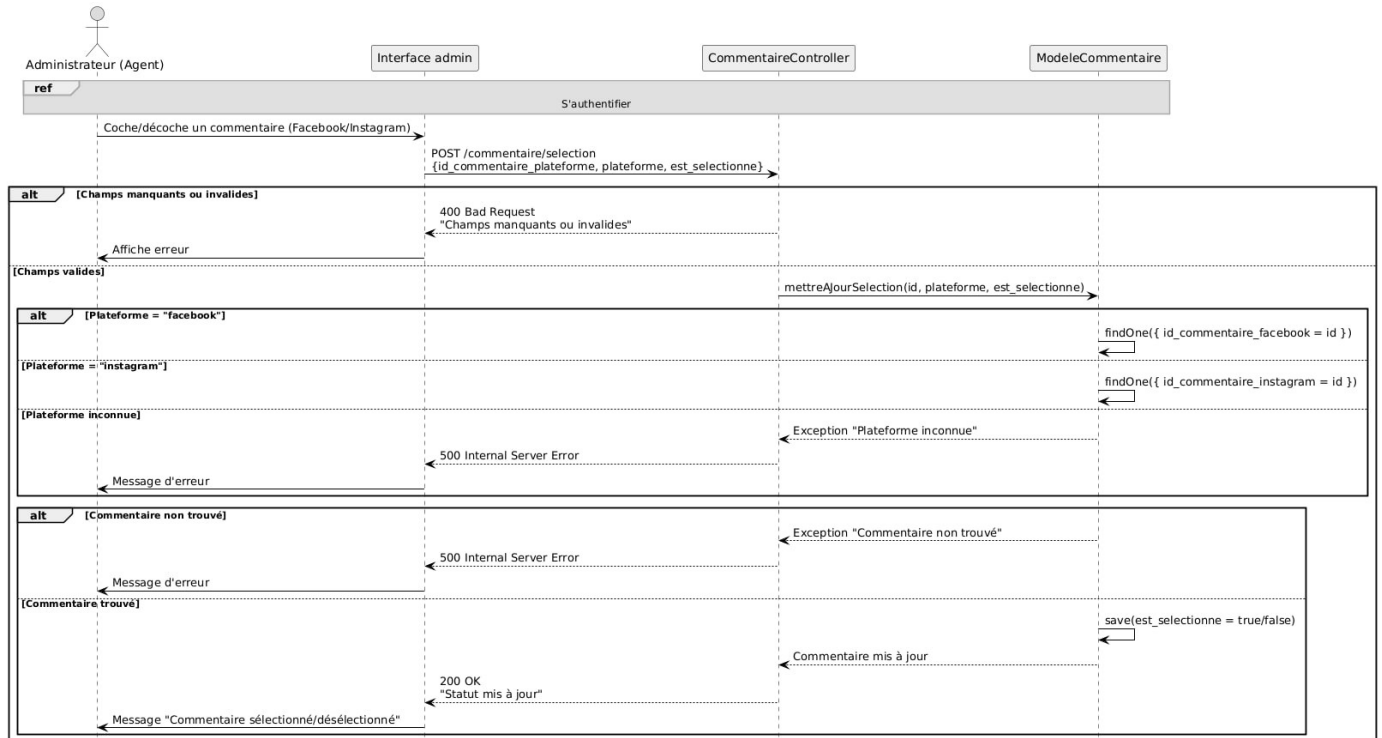


FIGURE 3.20 – Diagramme de séquence - Sélectionner des commentaires

### 3.3.7 Cas d'utilisation Réserver une Omra

#### 3.3.7.1 Description textuelle de cas d'utilisation Réserver une Omra

Le tableau 3.12 montre la description textuelle de cas d'utilisation « Réserver une Omra ».

<b>Cas d'Utilisation</b>	Réserver une Omra
<b>But</b>	Ce cas permet à un utilisateur inscrit de réserver une offre Omra disponible via l'interface frontale de l'application.
<b>Acteur principal</b>	Utilisateur inscrit
<b>Acteur secondaire</b>	/
<b>Préconditions</b>	<ul style="list-style-type: none"> <li>— L'utilisateur est inscrit et connecté à son compte.</li> <li>— Une ou plusieurs offres Omra sont disponibles dans la base de données.</li> </ul>

<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur accède à la section des Omras dans l'application.</li> <li>2. Le système (frontend) interroge la base de données pour récupérer toutes les offres.</li> <li>3. Le système affiche les offres Omra (disponibles et non disponibles).</li> <li>4. L'utilisateur clique sur une Omra spécifique pour consulter ses détails.</li> <li>5. Le système récupère et affiche les détails de l'offre.</li> <li>6. L'utilisateur clique sur le bouton "Réserver maintenant".</li> <li>7. Le système affiche un formulaire de réservation.</li> <li>8. L'utilisateur remplit les champs requis (nom, prénom, dates, passeport, etc.) et accepte les conditions générales.</li> <li>9. L'utilisateur clique sur "Confirmer la réservation".</li> <li>10. Le système vérifie que tous les champs requis sont valides.</li> <li>11. Le système vérifie que l'offre Omra est toujours disponible.</li> <li>12. Le système enregistre la réservation avec le statut "en attente".</li> <li>13. Le système affiche un message de succès et informe que la réservation est en attente de paiement.</li> </ol>
<b>Enchaînement alternatif</b>	<ul style="list-style-type: none"> <li>— Champs invalides ou manquants : Le système retourne un message d'erreur précisant les champs à corriger.</li> <li>— Offre Omra non disponible : Le système affiche un message d'erreur indiquant que l'offre n'est plus disponible.</li> </ul>
<b>Postcondition</b>	La réservation est enregistrée avec succès dans la base de données avec le statut "en attente".

TABLE 3.12 – Description textuelle du cas d'utilisation Réserver une Omra

### 3.3.7.2 Diagramme de séquence de Cas d'utilisation « Réserver une Omra »

La figure 3.21 montre le diagramme de séquence de cas d'utilisation « Réserver une Omra ».

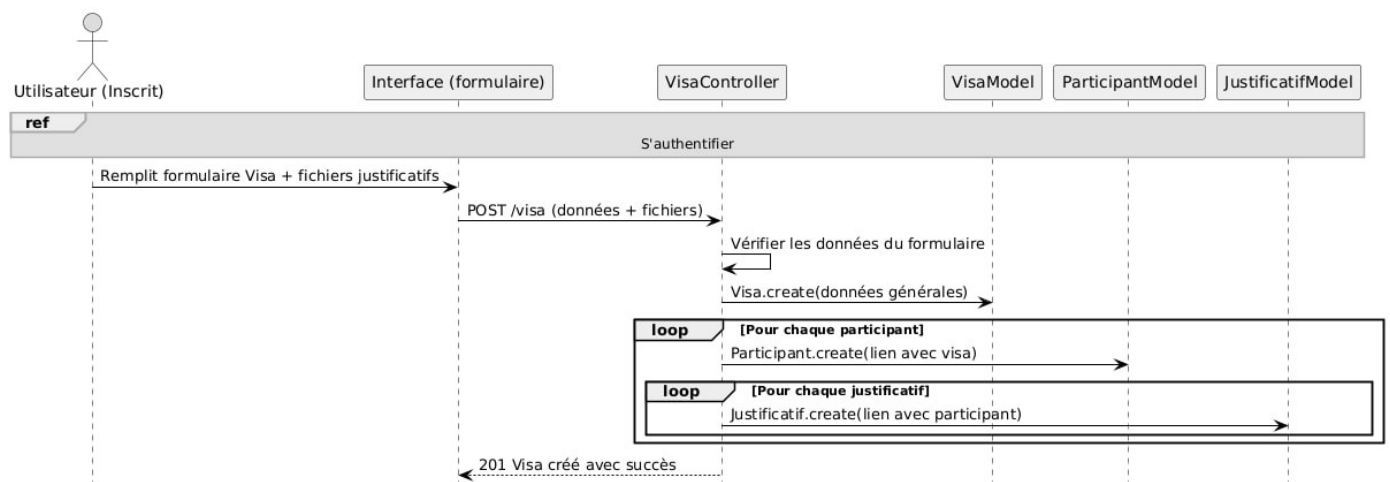


FIGURE 3.21 – Diagramme de séquence - Réserver une Omra

### **3.3.8 Cas d'utilisation Filtrer une Réserveation**

#### **3.3.8.1 Description textuelle de cas d'utilisation Filtrer une Réserveation**

Le tableau 3.13 montre la description textuelle de cas d'utilisation « Filtrer les réservations ».

<b>Cas d'Utilisation</b>	Filtrer les réservations
<b>But</b>	Ce cas permet à l'administrateur ou à l'agent de filtrer les réservations selon des critères (statut, date) pour faciliter la recherche et le suivi.
<b>Acteur principal</b>	Administrateur / Agent
<b>Acteur secondaire</b>	/
<b>Préconditions</b>	L'utilisateur est authentifié et a accédé à l'interface de gestion des réservations du back-office.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur (administrateur ou agent) se connecte au back-office.</li> <li>2. Le système affiche le tableau de bord.</li> <li>3. L'utilisateur clique sur "Réservations", puis sur "Liste des réservations".</li> <li>4. Le système affiche tous les enregistrements de réservation.</li> <li>5. L'utilisateur saisit un ou plusieurs critères de filtrage (ex : statut, date de réservation).</li> <li>6. Le système interroge la base de données avec les critères fournis.</li> <li>7. Le système affiche la liste des réservations correspondant aux filtres.</li> </ol>
<b>Enchaînement alternatif</b>	— Aucun résultat trouvé : Si aucune réservation ne correspond aux critères de filtrage, le système affiche un message : « Aucune réservation trouvée pour ces filtres. »
<b>Postcondition</b>	Si des résultats existent, ils sont affichés à l'utilisateur. Sinon, un message l'informe qu'aucune réservation ne correspond aux filtres.

TABLE 3.13 – Description textuelle du cas d'utilisation « Filtrer les réservations »

### 3.3.8.2 Diagramme de séquence de Cas d'utilisation Filtrer une réservation

La figure 3.22 montre le diagramme de séquence de cas d'utilisation « Filtrer une réservation ».

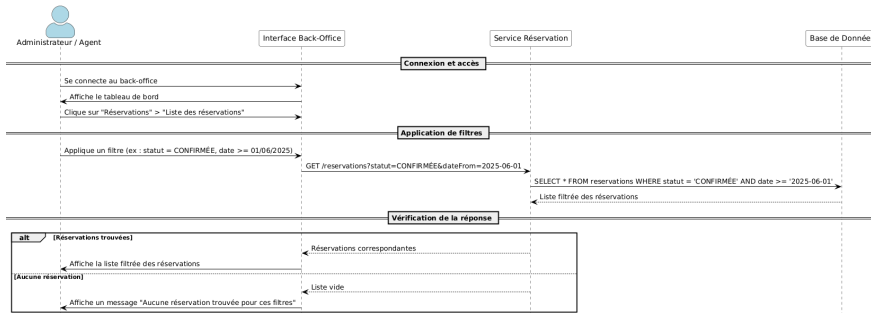


FIGURE 3.22 – Diagramme de séquence - Filtrer une réservation

### 3.3.9 Cas d'utilisation Rechercher un vol

#### 3.3.9.1 Description textuelle de cas d'utilisation Rechercher un vo

Le tableau 3.14 montre la description textuelle de cas d'utilisation « Rechercher un vol ».

<b>Cas d'Utilisation</b>	Rechercher un vol
<b>But</b>	Ce cas permet à un utilisateur inscrit d'effectuer une recherche de vols en fonction de critères comme l'origine, la destination, les dates, le nombre de passagers et la classe.
<b>Acteur principal</b>	Utilisateur inscrit
<b>Acteur secondaire</b>	API Amadeus
<b>Préconditions</b>	L'utilisateur est inscrit, connecté, et a accédé à la page de recherche de vols.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur accède à la page de recherche de vol depuis l'interface front-end.</li> <li>2. Le système affiche un formulaire de recherche.</li> <li>3. L'utilisateur saisit les informations nécessaires : origine, destination, dates de départ et de retour, nombre de passagers, classe.</li> <li>4. Le système envoie la requête au backend.</li> <li>5. Le backend vérifie la validité des champs saisis (dates, lieux, etc.).</li> <li>6. Si les champs sont valides, le système envoie une requête à l'API Amadeus avec les paramètres de recherche.</li> <li>7. L'API retourne une liste de vols correspondant aux critères.</li> <li>8. Le système affiche à l'utilisateur les vols disponibles avec leurs détails.</li> </ol>
<b>Enchaînement alternatif</b>	<ul style="list-style-type: none"> <li>— Date invalide ou déjà passée : Le système affiche un message d'erreur.</li> <li>— Aucun vol trouvé : Le système informe l'utilisateur qu'aucun vol ne correspond aux critères saisis.</li> </ul>
<b>Postcondition</b>	Les résultats de la recherche sont affichés à l'utilisateur ou un message approprié est présenté si aucune correspondance n'a été trouvée.

TABLE 3.14 – Description textuelle du cas d'utilisation « Rechercher un vol »

### 3.3.9.2 Diagramme de séquence de Cas d'utilisation rechercher un vol

La figure 3.23 montre le diagramme de séquence de cas d'utilisation « Rechercher un vol ».

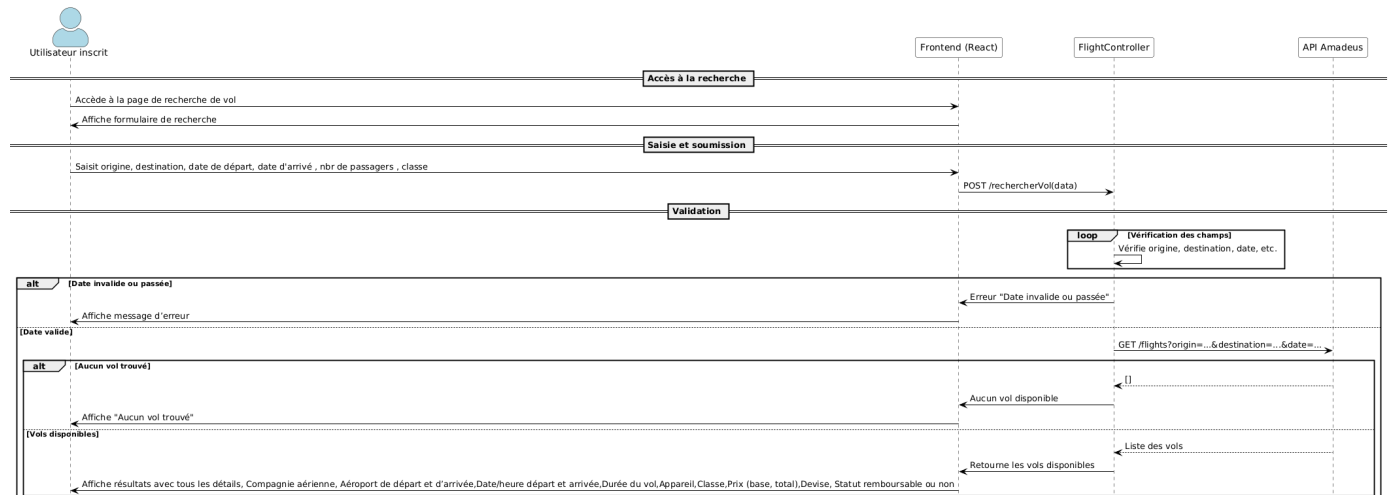


FIGURE 3.23 – Diagramme de séquence - Rechercher un vol

### 3.3.10 Cas d'utilisation Rechercher un Hôtel

#### 3.3.10.1 Description textuelle de cas d'utilisation Rechercher un hôtel

Le tableau 3.15 montre la description textuelle de cas d'utilisation « Rechercher un Hôtel ».

<b>Cas d'Utilisation</b>	Rechercher un hôtel
<b>But</b>	Ce cas permet à un utilisateur de rechercher une liste d'hôtels selon une ville, des dates et des critères de réservation, en récupérant les données depuis Booking.com.
<b>Acteur principal</b>	Utilisateur
<b>Acteur secondaire</b>	Interface front-end, HotelController, API Booking.com
<b>Préconditions</b>	L'utilisateur accède à l'interface de recherche d'hôtels. L'API Booking.com est accessible et opérationnelle.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur accède à la page de recherche d'hôtels.</li> <li>2. Il saisit les critères de recherche : ville, dates, nombre de chambres, adultes, enfants.</li> <li>3. L'interface front-end envoie une requête GET au serveur avec les paramètres de recherche.</li> <li>4. Le contrôleur (HotelController) transmet la requête à l'API Booking.com.</li> <li>5. L'API retourne une liste d'hôtels correspondant aux critères.</li> <li>6. Le contrôleur renvoie la réponse (code 200 + données JSON) à l'interface.</li> <li>7. L'interface affiche la liste des hôtels disponibles à l'utilisateur.</li> </ol>
<b>Enchaînement alternatif</b>	— Erreur API : Si l'API Booking.com retourne une erreur, le contrôleur capture l'erreur et renvoie une réponse 500 avec un message « Erreur serveur ».
<b>Postcondition</b>	La liste des hôtels est affichée si la requête réussit. Sinon, un message d'erreur est présenté à l'utilisateur.

TABLE 3.15 – Description textuelle du cas d'utilisation « Rechercher un hôtel »

### 3.3.10.2 Diagramme de séquence de Cas d'utilisation Rechercher un hôtel

La figure 3.24 montre le diagramme de séquence de cas d'utilisation « Rechercher un hôtel ».

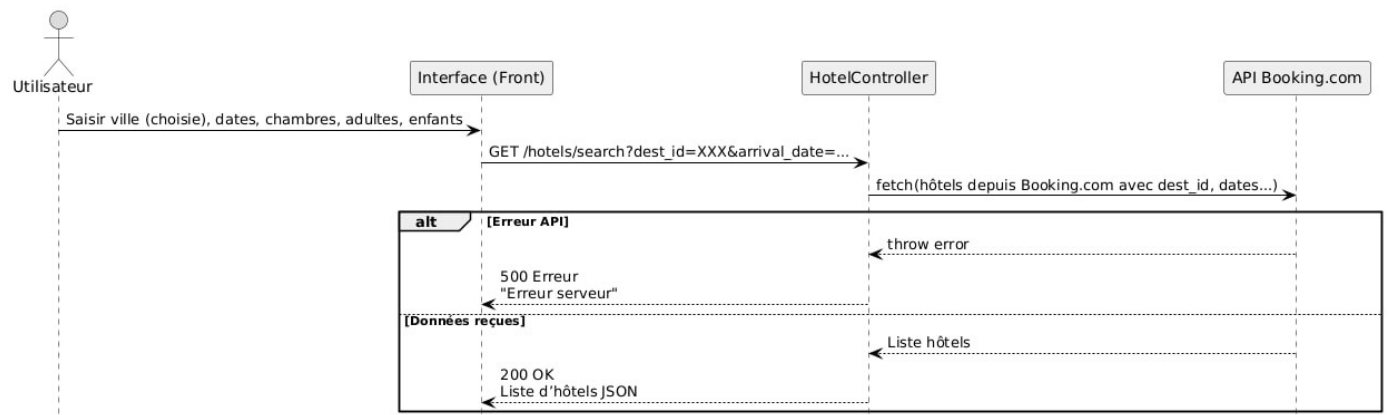


FIGURE 3.24 – Diagramme de séquence - Rechercher un hôtel

### 3.3.11 Cas d'utilisation Payer avec Chargily

#### 3.3.11.1 Description textuelle de cas d'utilisation Payer avec chargily

Le tableau 3.16 montre la description textuelle de cas d'utilisation « Payer avec chargily ».

<b>Cas d'Utilisation</b>	Payer avec Chargily
<b>But</b>	Ce cas permet à un utilisateur inscrit de régler sa réservation via la passerelle de paiement Chargily ePay en utilisant une carte CIB ou Edahabia.
<b>Acteur principal</b>	Utilisateur inscrit
<b>Acteur secondaire</b>	API Chargily ePay
<b>Préconditions</b>	L'utilisateur est inscrit, connecté, et dispose d'une réservation valide non encore payée.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur accède à l'interface de paiement.</li> <li>2. Le système affiche les différentes options de paiement.</li> <li>3. L'utilisateur choisit le mode "Carte CIB/Edahabia (via Chargily)".</li> <li>4. Le frontend envoie la requête au backend avec l'identifiant de la réservation.</li> <li>5. Le backend vérifie l'ID de réservation.</li> <li>6. Si l'identifiant est valide, le backend envoie une requête à l'API Chargily pour générer un lien de paiement.</li> <li>7. L'API retourne un lien de paiement avec un statut "en attente".</li> <li>8. Le backend transmet ce lien au frontend.</li> <li>9. L'utilisateur est redirigé vers l'interface de paiement Chargily.</li> <li>10. L'utilisateur saisit les informations nécessaires et valide le paiement.</li> <li>11. L'interface Chargily affiche un message de confirmation.</li> <li>12. Le serveur reçoit une notification automatique via le webhook Chargily.</li> <li>13. Le webhook met à jour la base de données : le statut du paiement devient "payé", la réservation est "confirmée", et le rôle de l'utilisateur devient "client".</li> </ol>
<b>Enchaînements alternatifs</b>	<ul style="list-style-type: none"> <li>— <b>Réservation introuvable</b> : Le backend retourne une erreur et le frontend affiche un message d'erreur à l'utilisateur.</li> <li>— <b>Échec du paiement</b> : Si Chargily ne renvoie pas de confirmation, la réservation reste en attente.</li> </ul>
<b>Postcondition</b>	Si le paiement est validé, la réservation est confirmée et le rôle de l'utilisateur est mis à jour.

TABLE 3.16 – Description textuelle du cas d'utilisation « Payer avec Chargily »

### 3.3.11.2 Diagramme de séquence de Cas d'utilisation Payer avec Chargily

La figure 3.25 montre le diagramme de séquence de cas d'utilisation « Payer avec Chargily ».

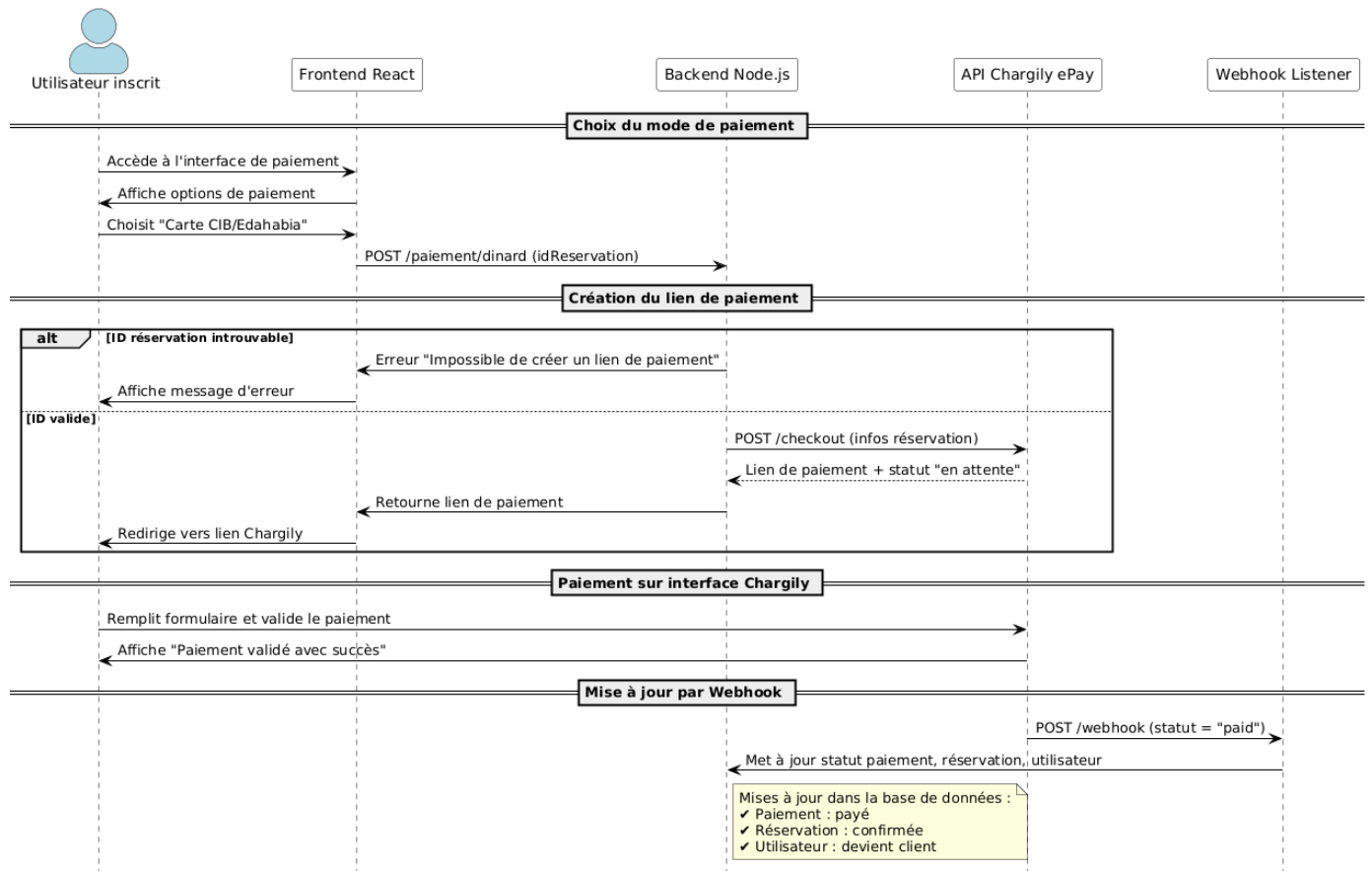


FIGURE 3.25 – Diagramme de séquence - Payer avec Chargily

### 3.3.12 Cas d'utilisation Envoyer les factures par email

#### 3.3.12.1 Description textuelle de cas d'utilisation

Le tableau 3.17 montre la description textuelle de cas d'utilisation « Envoyer factures par email ».

<b>Cas d'Utilisation</b>	Envoyer facture par email
<b>But</b>	Ce cas permet d'envoyer automatiquement la facture de paiement au format PDF par email à l'utilisateur après la confirmation de son paiement.
<b>Acteur principal</b>	Utilisateur inscrit
<b>Acteurs secondaires</b>	API Chargily ePay / Stripe, Service Email
<b>Préconditions</b>	Le paiement a été effectué et confirmé par la plateforme de paiement (Chargily ou Stripe).
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'API Chargily ou Stripe notifie le serveur via un webhook que le paiement est confirmé (statut = "paid").</li> <li>2. Le webhook interroge le backend pour récupérer les informations liées à la réservation et à l'utilisateur.</li> <li>3. Le backend génère un identifiant unique pour la facture.</li> <li>4. Le backend regroupe toutes les informations nécessaires concernant l'utilisateur.</li> <li>5. Le backend ajoute les détails de la réservation associée.</li> <li>6. Le backend définit le statut de la facture à "payé".</li> <li>7. Le backend génère un fichier PDF contenant la facture complète.</li> <li>8. Le backend envoie un email à l'utilisateur avec la facture PDF en pièce jointe via le service email.</li> <li>9. L'utilisateur reçoit l'email contenant la facture.</li> </ol>
<b>Enchaînement alternatif</b>	Aucun
<b>Postcondition</b>	L'utilisateur reçoit par email une facture au format PDF correspondant au paiement confirmé.

TABLE 3.17 – Description textuelle du cas d'utilisation « Envoyer une facture par email »

### 3.3.12.2 Diagramme de séquence de Cas d'utilisation Envoyer factures par email

La figure 3.26 montre le diagramme de séquence de cas d'utilisation « Envoyer factures par email ».

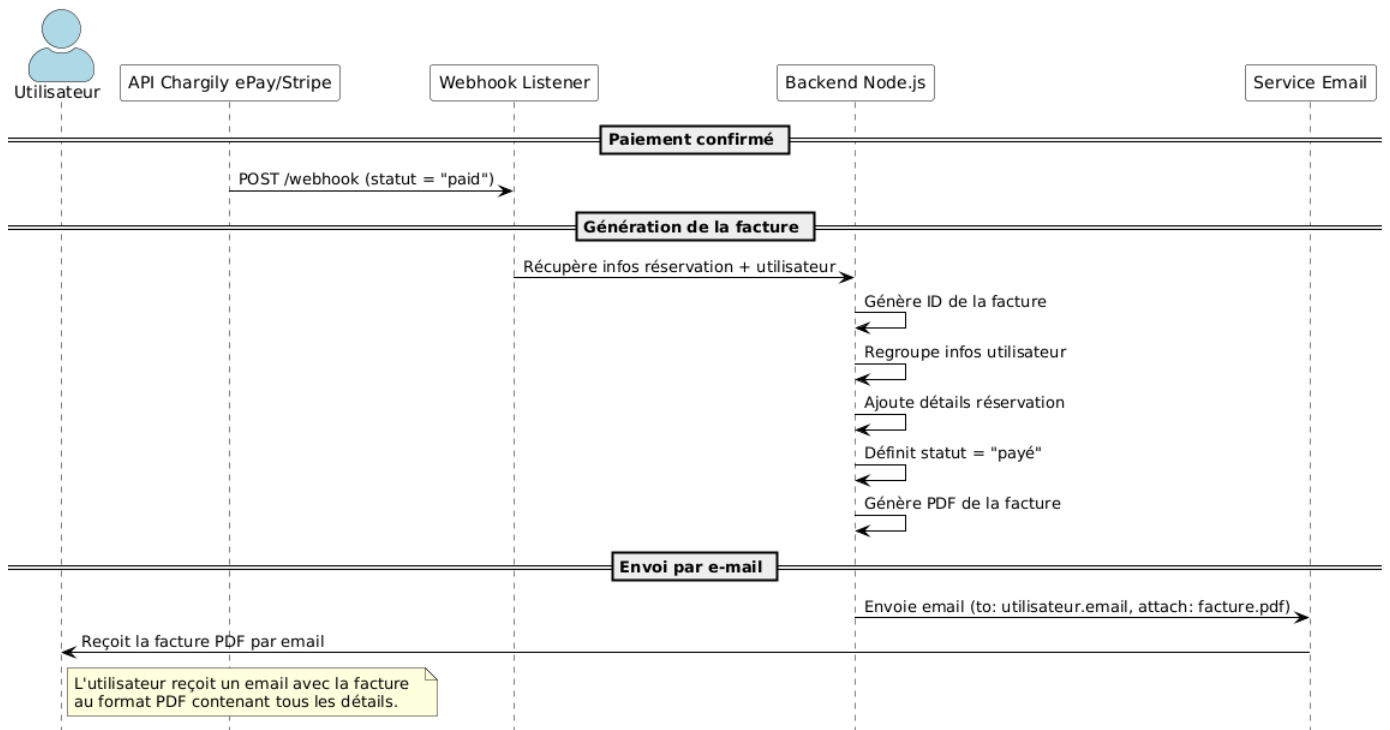


FIGURE 3.26 – Diagramme de séquence - Envoyer factures par email

### 3.3.13 Cas d'utilisation Configurer les APIs

#### 3.3.13.1 Description textuelle de cas d'utilisation Configurer les APIs

Le tableau 3.18 montre la description textuelle de cas d'utilisation « Configurer les apis ».

<b>Cas d'Utilisation</b>	Configurer les APIs
<b>But</b>	Permettre à l'administrateur de gérer et configurer les clés API et tokens des services tiers (ex : Chargily, Stripe, Amadeus) et passer en mode production (live).
<b>Acteur principal</b>	Administrateur
<b>Acteurs secondaires</b>	Aucun
<b>Préconditions</b>	L'administrateur est connecté au back-office avec les droits nécessaires.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'administrateur se connecte au back-office.</li> <li>2. Le back-office affiche le tableau de bord.</li> <li>3. L'administrateur accède à la section "Configuration des API".</li> <li>4. Le back-office demande la liste des APIs au service de configuration.</li> <li>5. Le service interroge la base de données et récupère les clés, tokens, etc.</li> <li>6. La base renvoie les données au service.</li> <li>7. Le service transmet la liste des APIs au back-office.</li> <li>8. Le back-office affiche les informations à l'administrateur.</li> <li>9. L'administrateur modifie une clé API (ex : passage en mode live).</li> <li>10. Le back-office envoie la mise à jour au service.</li> <li>11. Le service met à jour la base de données.</li> <li>12. La base confirme la mise à jour.</li> <li>13. Le service informe le back-office.</li> <li>14. Le back-office affiche une confirmation à l'administrateur.</li> </ol>
<b>Enchaînement alternatif</b>	En cas d'erreur lors de la récupération ou de la mise à jour des clés API, un message d'erreur est affiché à l'administrateur.
<b>Postcondition</b>	Les clés API et tokens sont correctement configurés et mis à jour dans la base de données.

TABLE 3.18 – Description textuelle du cas d'utilisation « Configurer les APIs »

### 3.3.13.2 Diagramme de séquence de Cas d'utilisation Configurer les APIs

La figure 3.27 le diagramme de séquence de cas d'utilisation « Configurer les apis ».

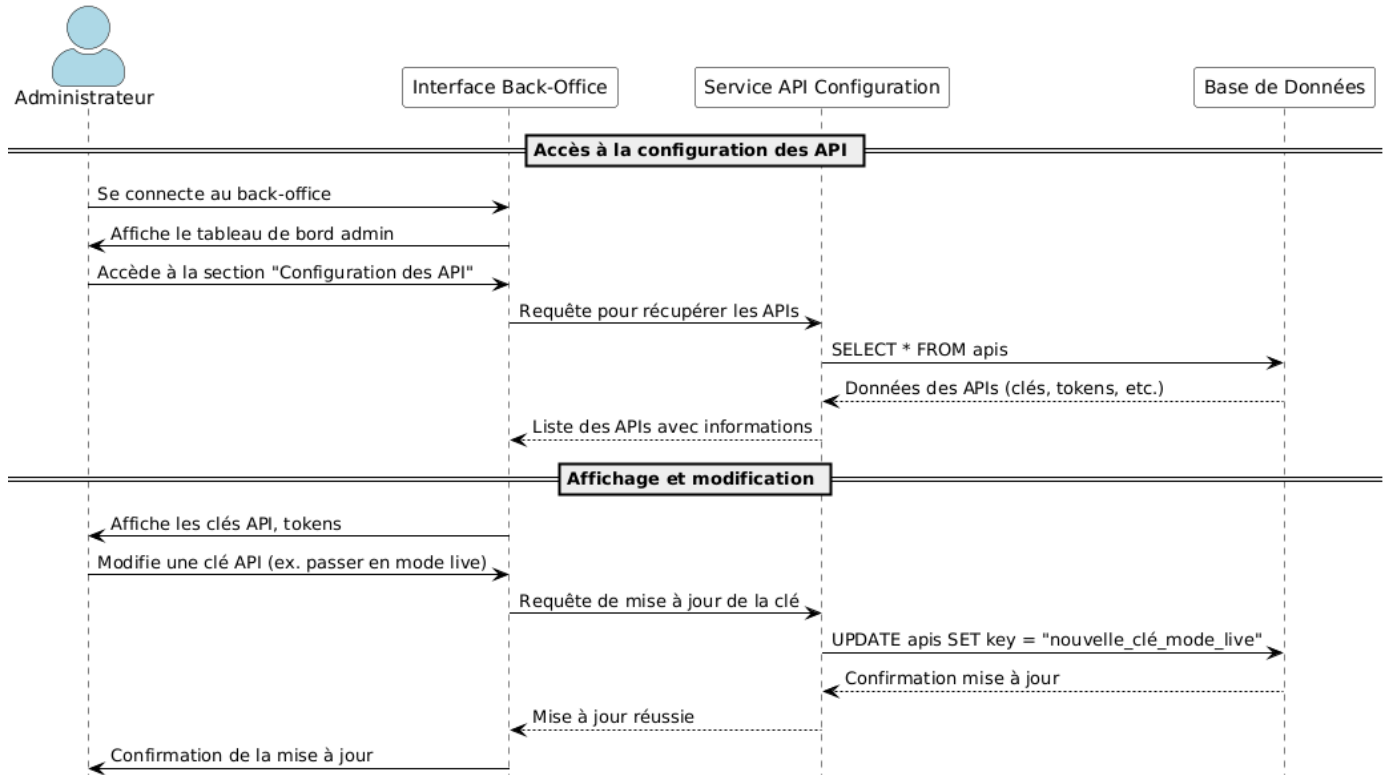


FIGURE 3.27 – Diagramme de séquence - Configurer les APIs

### 3.3.13.3 Diagramme de Classe deuxième Sprint

La figure 3.28 le diagramme de classe de deuxième sprint.

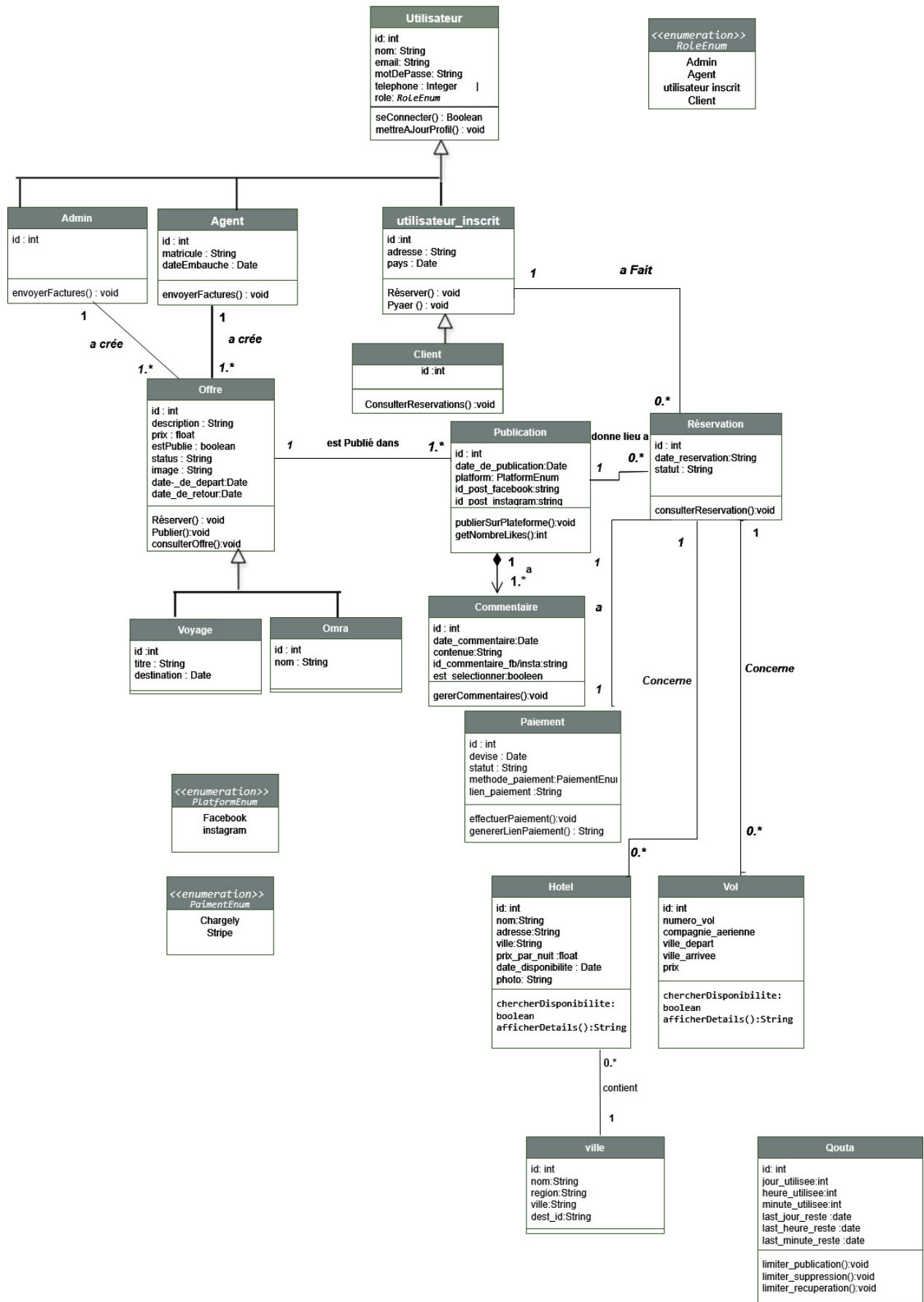


FIGURE 3.28 – Diagramme de classe - Deuxième sprint

### 3.3.14 Modèle relationnel du 2<sup>ème</sup> sprint

Voici le modèle relationnel correspondant au diagramme de classes du deuxième sprint.

**Client**(id) – id est à la fois PK et FK vers Utilisateur\_inscrit

**Voyage**(id, titre, description, date\_depart, destination, prix, statut, est\_publie, image, #id\_agent,#id\_admin)

**Omra**(id, nom, description, prix, date\_depart, duree, est\_publie, statut, image, #id\_agent,#id\_admin)

**Publication**(id, date\_publication, plateforme, statut, #id\_voyage, #id\_omra, id\_post\_facebook, id\_post\_instagram)

**Commentaire**(id, date\_commentaire, contenu, #id\_publication)

**Reservation**(id, date\_reservation, statut, #id\_utilisateur\_inscrit, #id\_publication, #id\_vol, #id\_hotel)

**Paiement**(id, devise, methode\_paiement, statut, lien\_paiement, #id\_reservation)

**Vol**(id, numero\_vol, compagnie\_aerienne, ville\_depart, ville\_arrivee, prix)

**Hotel**(id, nom, adresse, ville, prix\_par\_nuit, date\_disponibilite,photo, #id\_ville)

**Ville**(id, nom, region, dest\_id)

**Quota**(id, jour\_utilise, heure\_utilise, minute\_utilise,last\_jour\_reste, last\_heure\_reste, last\_minuteur\_reste)

### 3.3.15 Interfaces du deuxième sprint

#### 3.3.15.1 Interface Créer un voyage

L'agent/admin peut créer un voyage organisé en remplissant un formulaire détaillé.

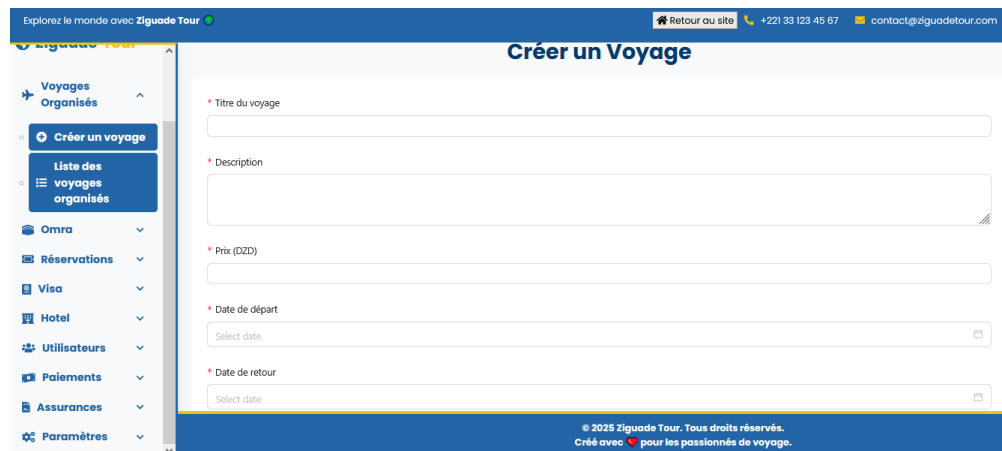


FIGURE 3.29 – Interface de création d'un voyage

#### 3.3.15.2 Interface Publier un voyage

Depuis le back-office, un agent ou un administrateur peut publier une offre de voyage directement sur la page Facebook, le compte Instagram, ainsi que sur le site web de l'agence.

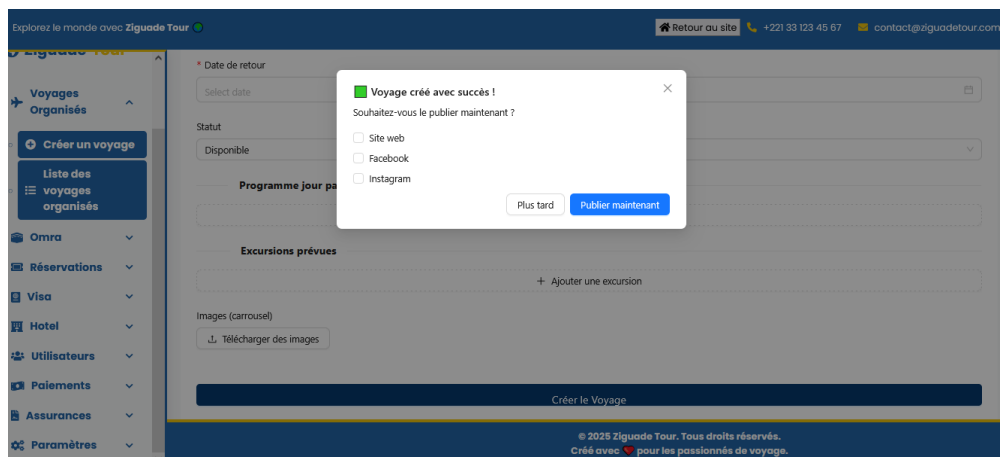


FIGURE 3.30 – Interface de publication d'un voyage

### 3.3.15.3 Interface liste de publication de chaque offre

L'agent ou l'administrateur peut consulter la liste des publications associées à chaque offre (voyage ou Omra), afin de vérifier sur quels canaux celle-ci a été publiée (site web, Facebook, Instagram).

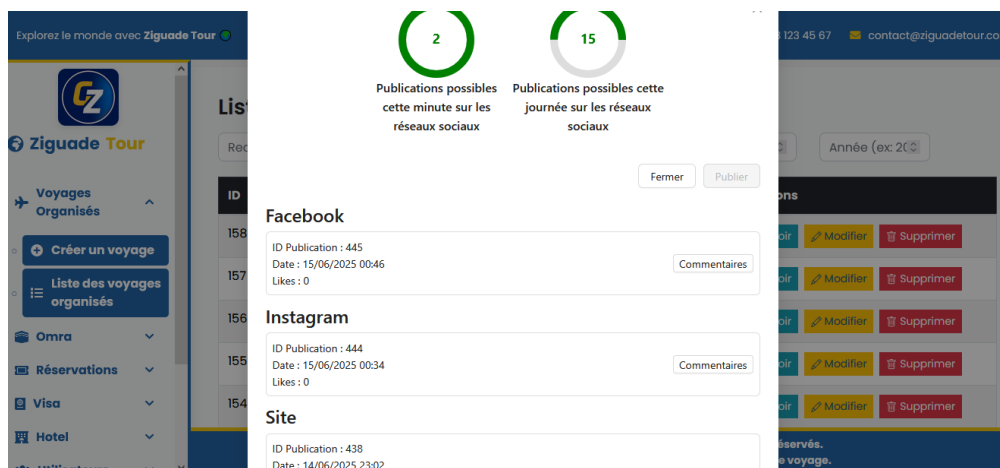


FIGURE 3.31 – Interface liste de publication pour chaque offre

### 3.3.15.4 Interface Récupération des commentaires de chaque publication

L'interface permet de récupérer automatiquement les (anciens/ nouveaux) commentaires publiés sur les publications Facebook et Instagram et pouvoir les supprimer ou les sélectionner.

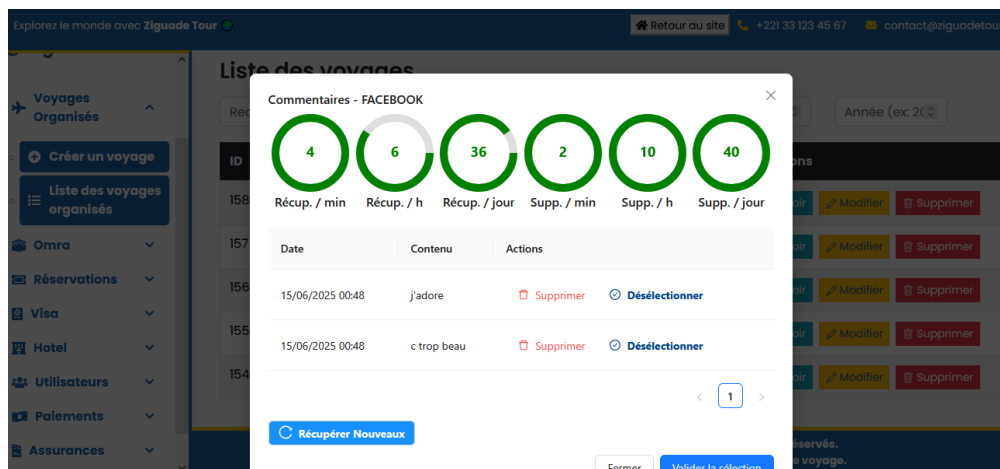


FIGURE 3.32 – Interface de récupération des commentaires

### 3.3.15.5 Interface publication de voyage sur le site

L'utilisateur peut voir les détails d'un voyage publié sur le site.

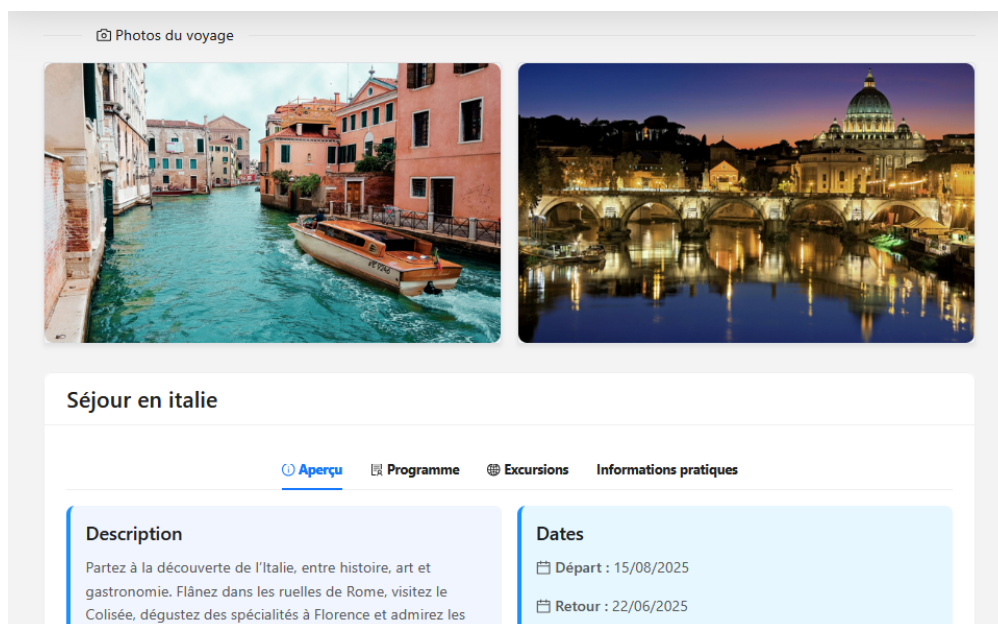


FIGURE 3.33 – Interface de publication de voyage sur le site

### 3.3.15.6 Interface publication omra sur le site

L'utilisateur peut voir les détails d'une Omra publié sur le site.

**Programme**

La Omrah est un pèlerinage spirituel important pour les musulmans, effectué à La Mecque. Les rituels incluent Tawaf autour de la Kaaba, Sa'i entre Safa et Marwah, et prières de Tarawih pour renforcer foi et communauté.

**Jour 1 : Arrivée à Médina**

- **Heure** : Arrivée à l'aéroport de Médina.
- **Transfert** : Transfert à l'hôtel Arjowan Rose.
- **Matinée** : Repos à l'hôtel après le voyage.
- **Après-midi** : Visite de la Mosquée du Prophète (Masjid al-Nabawi).
- **Soirée** : Dîner à l'hôtel ou dans un restaurant local.

**Jour 2 : Médina**

- **Matin** : Prière à la Mosquée du Prophète, visite de la tombe du Prophète Muhammad (paix soit sur lui).
- **Après-midi** : Visite de sites historiques (Mosquée Quba, Montagne Uhud).
- **Soir** : Temps libre pour explorer les marchés locaux.

**Jour 3 : Médina**

- **Matin** : Prière à la Mosquée du Prophète.
- **Après-midi** : Visite de la mosquée Qiblatain et du cimetière de Baqi.
- **Soir** : Dîner et temps libre.

**Jour 4 : Départ pour La Mecque**






FIGURE 3.34 – Interface de publication de voyage sur le site

### 3.3.15.7 Capture Publication de voyage sur facebook

Le visiteur peut consulter les offres de voyages ou d'Omra publiées sur Facebook, et accéder à la même publication sur le site via un lien, afin de procéder à la réservation.



**Agence ziguede tour**  
Publié par ziguede Tour  
· 28 min ·

Nouveau voyage : Voyage en Espagne  
Description: Partez à la découverte des merveilles de l'Espagne : entre l'architecture de Gaudí à Barcelone, les palais andalous de l'Alhambra, et l'ambiance des ruelles animées de Madrid. Un séjour riche en patrimoine, cuisine, et traditions.  
Prix : 95000 €  
Date de départ : 2025-09-15  
Date de retour : 2025-09-22  
Réservez votre place ici : <https://1ed6-154-248-34-163.ngrok-free.app/.../infos.../157>



FIGURE 3.35 – Interface de capture de la publication sur facebook

### 3.3.15.8 Capture Publication de voyage sur instagram

Le visiteur peut consulter les offres de voyages ou d'Omra publiées sur Instagram, et accéder à la même publication sur le site via un lien, afin de procéder à la réservation.

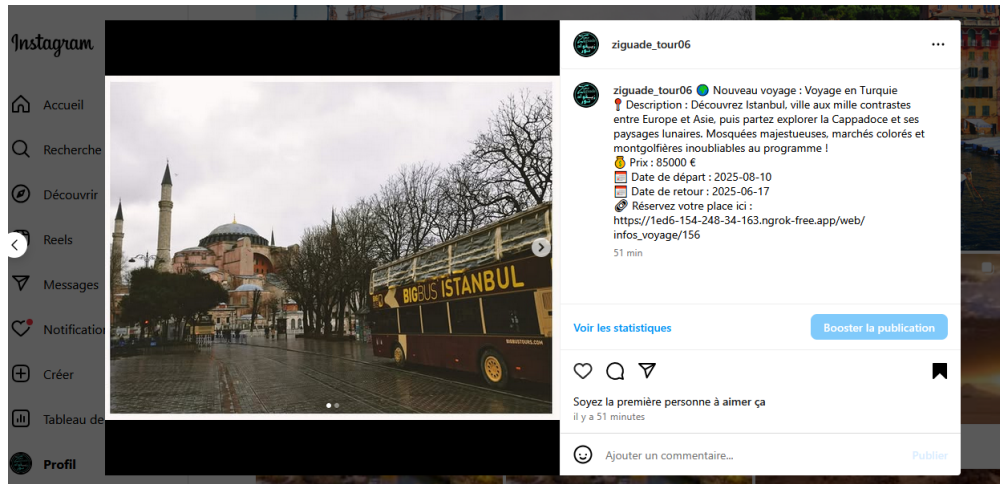


FIGURE 3.36 – Interface de capture de la publication sur instagram

### 3.3.15.9 Interface liste de voyages sur site avec commentaire sélectionner et nombre de like

Cette interface permet à un utilisateur inscrit ou à un simple visiteur de consulter la liste des voyages publiés sur le site. Pour chaque voyage, quelques commentaires représentatifs sont affichés. Ces derniers sont cliquables et redirigent directement vers la publication correspondante sur Facebook ou Instagram. L'utilisateur peut également voir le nombre total de mentions "J'aime" associées à chaque publication.

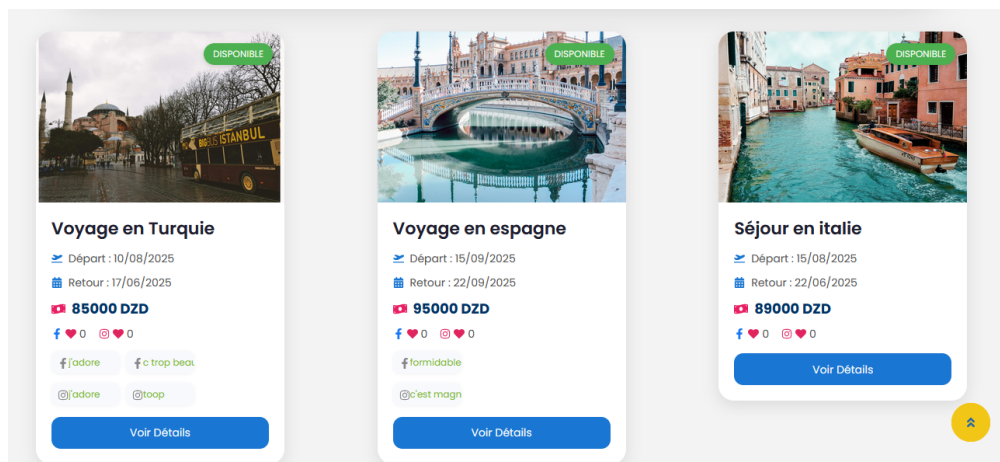


FIGURE 3.37 – Interface liste de voyage sur site avec commentaire sélectionner et nombre de likes

### 3.3.15.10 Interface liste d'omras sur site avec commentaire sélectionner et nombre de likes

Cette interface permet à un utilisateur inscrit ou à un simple visiteur de consulter la liste d'omras publiés sur le site. Pour chaque voyage, quelques commentaires représentatifs sont affichés. Ces derniers sont cliquables et redirigent directement vers la publication correspondante sur Facebook ou Instagram. L'utilisateur peut également voir le nombre total de mentions "J'aime" associées à chaque publication.

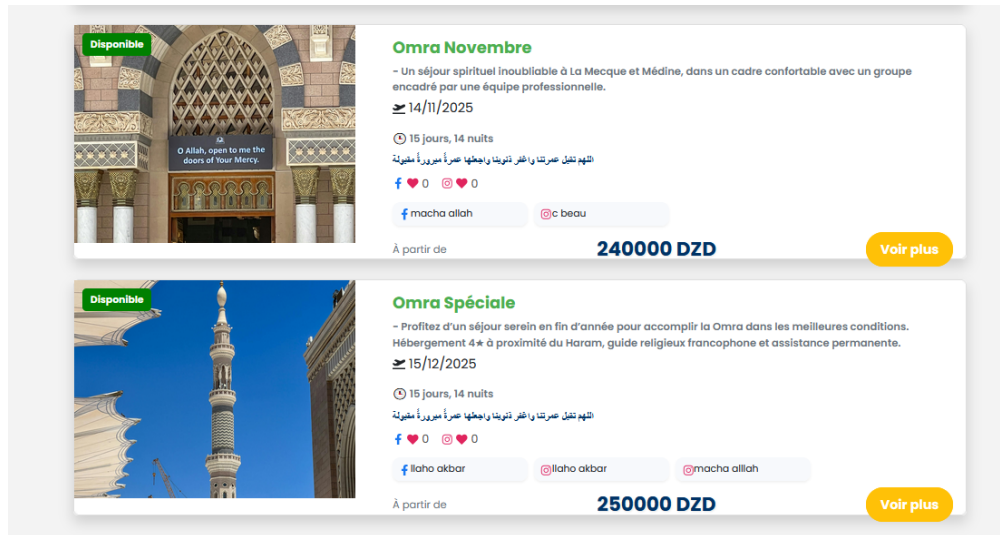


FIGURE 3.38 – Interface liste de voyage sur site avec commentaire sélectionner et nombre de likes

### 3.3.15.11 Interface Filtrer les réservations

Les agents peuvent utiliser des filtres pour trier les réservations selon la date ou le statut (en attente, confirmée, annulée).

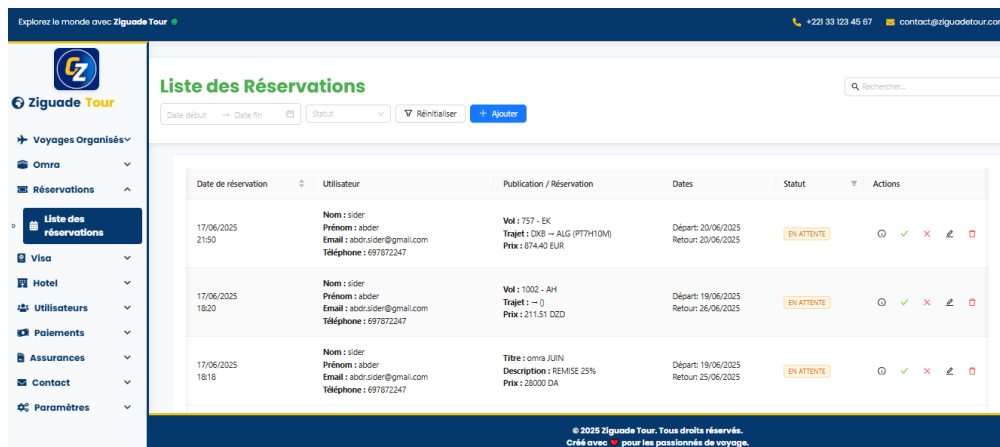


FIGURE 3.39 – Interface Filtrer les réservations

### 3.3.15.12 Interface Rechercher un vol

L'utilisateur inscrit peut rechercher un vol en précisant les aéroports de départ et d'arrivée, les dates, la class et le nombre de passagers.

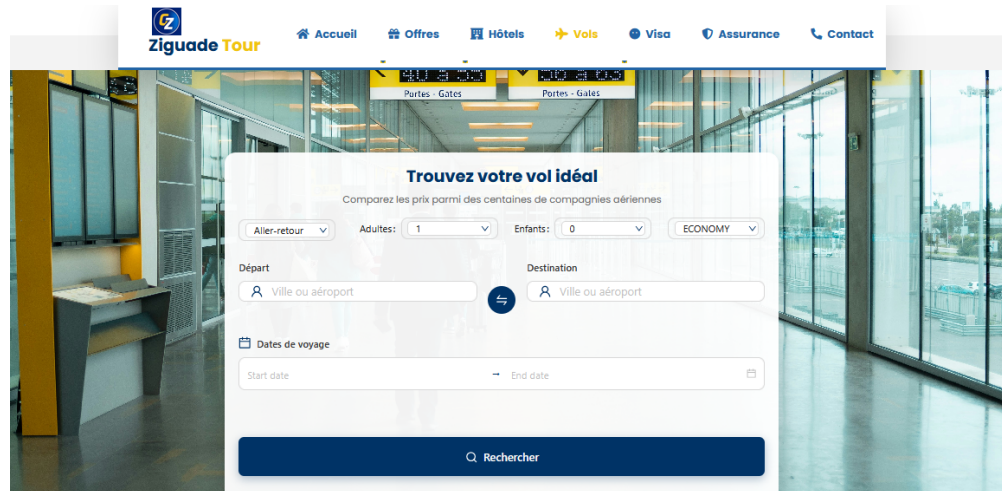


FIGURE 3.40 – Interface de recherche de vols

### 3.3.15.13 Interface Rechercher un hôtel

L'utilisateur inscrit peut rechercher un hôtel en sélectionnant la ville, la période et le nombre de personnes.

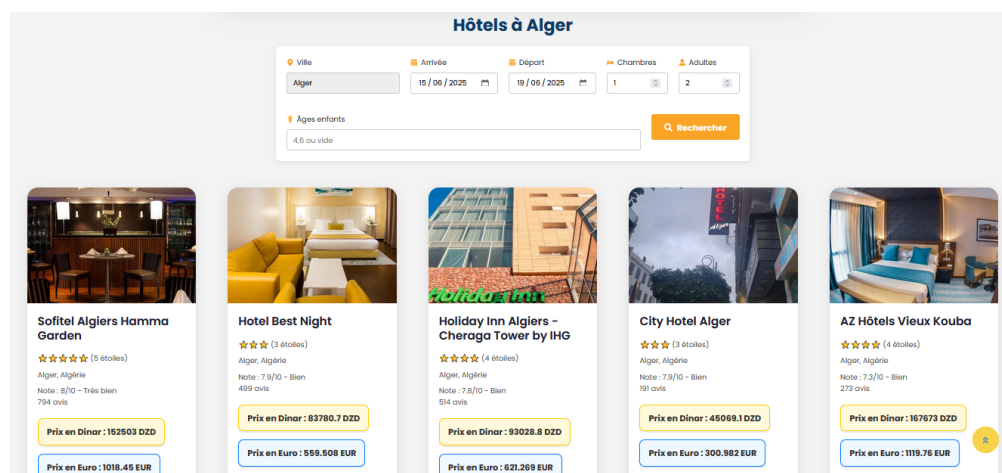


FIGURE 3.41 – Interface de recherche d'hôtels

### 3.3.15.14 Interface de choix de paiement

Lorsqu'un utilisateur inscrit finalise une réservation, il peut procéder au paiement sécurisé en ligne via plusieurs choix .

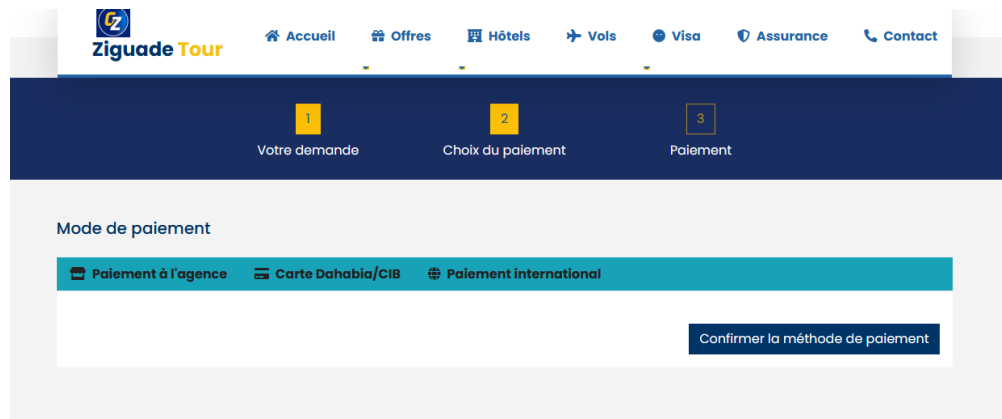


FIGURE 3.42 – interface choix de paiement

### 3.3.15.15 Interface Paiement avec Chargily ePay

Lorsqu'un utilisateur inscrit finalise une réservation, il peut procéder au paiement sécurisé en ligne via Chargily ePay.

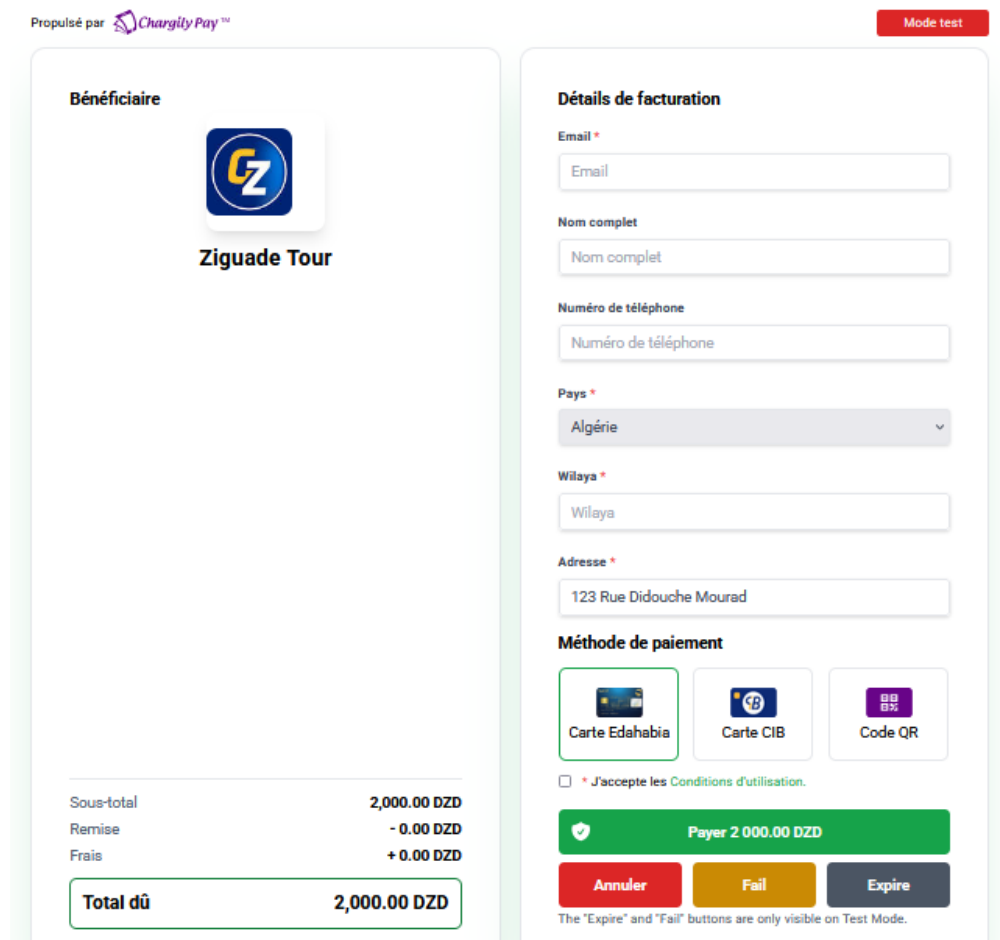


FIGURE 3.43 – Interface de paiement via Chargily

### 3.3.15.16 Interface Paiement avec Stripe

Les utilisateurs étrangers peuvent effectuer leurs paiements via la passerelle de paiement Stripe.

← ZIGUADE TEST MODE

Vol de à  
**913,15 \$US**

Payer avec link

Ou

E-mail abdr.sider@gmail.com

Informations de la carte

1234 1234 1234 1234

MM / AA CVC

Nom du titulaire de la carte

Nom complet

Pays ou région

Algérie

Enregistrer mes informations en toute sécurité pour pouvoir payer mes prochains achats en un clic  
Réglez plus rapidement vos achats auprès de ZIGUADE et de tous les autres professionnels qui acceptent Link.

Payer

Propulsé par stripe

Conditions d'utilisation Confidentialité

FIGURE 3.44 – Interface de paiement avec Stripe

### 3.3.15.17 Interface Mes factures

Le client peut télécharger sa facture de paiement



FIGURE 3.45 – Interface Mes factures

### 3.3.15.18 Interface Configuration des APIs

L'administrateur peut accéder à un espace dédié à la configuration des APIs externes. Il peut y ajouter les clés de Chargily, Stripe, ou Amadeus, et basculer en mode (test ou live).

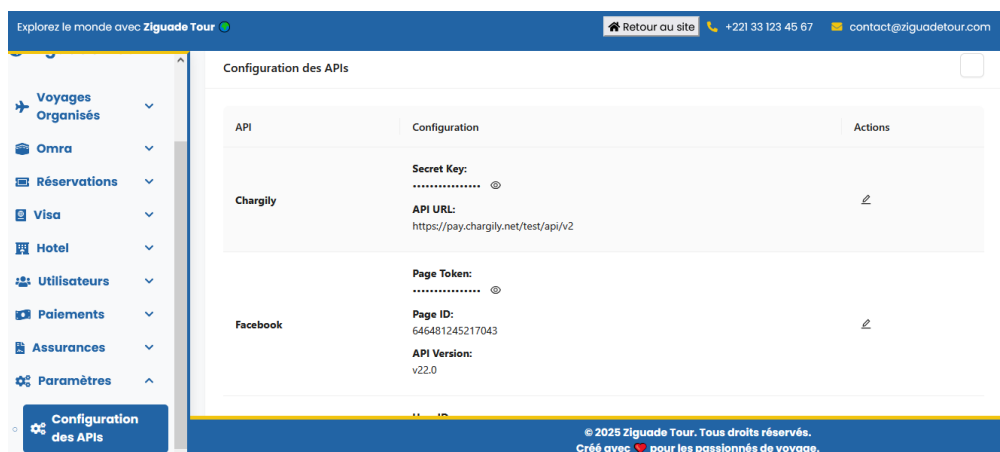


FIGURE 3.46 – Interface de configuration des APIs

## 3.4 Étude de 3ème Sprint

On estime que ce premier sprint va durer 6 jours et se compose de 3 items tels que :

- Gérer les visas
- Gérer les assurances
- Gérer la messagerie

### 3.4.1 Décomposition de Sprint 3

Tout au long de cette période, nous travaillons sur toutes les tâches de ce sprint afin de générer un incrément livrable. La figure illustre la répartition temporelle du sprint 3 en termes de chronologie.

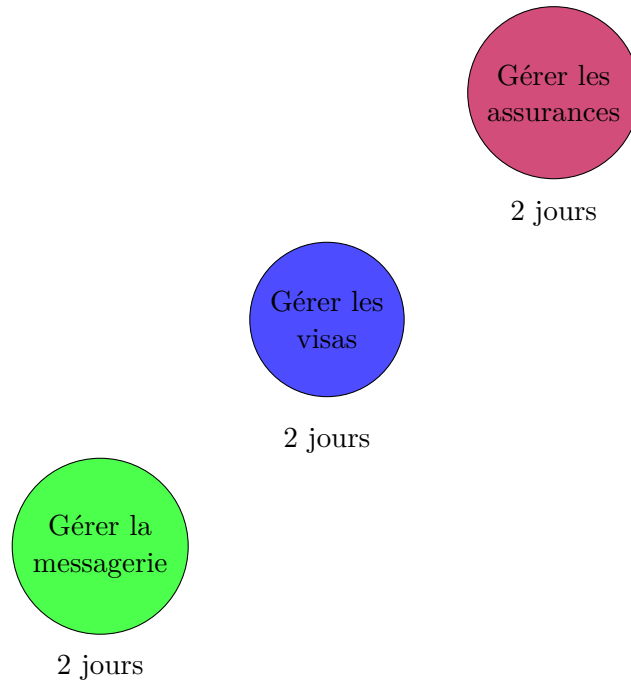


FIGURE 3.47 – Répartition temporelle du sprint 3

### 3.4.2 Diagramme de cas d'utilisation Globale Sprint 3

La figure 3.48 montre le diagramme d'utilisation du 3ème sprint.



FIGURE 3.48 – Diagramme de cas d'utilisation - 3ème sprint

### 3.4.3 Cas d'utilisation Souscrire une assurance

#### 3.4.3.1 Description textuelle de cas d'utilisation

Le tableau 3.19 montre la description textuelle de cas d'utilisation « Souscrire une assurance ».

<b>Cas d'Utilisation</b>	Souscrire une assurance
<b>But</b>	Permettre à un utilisateur inscrit de souscrire une assurance via un formulaire en ligne.
<b>Acteur principal</b>	Utilisateur inscrit
<b>Acteurs secondaires</b>	Aucun
<b>Préconditions</b>	L'utilisateur est connecté et possède un compte valide.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur accède à la page Assurance.</li> <li>2. L'interface affiche un formulaire de souscription.</li> <li>3. L'utilisateur remplit le formulaire et choisit le type d'assurance.</li> <li>4. L'utilisateur clique sur "Souscrire une assurance".</li> <li>5. Le formulaire est envoyé au contrôleur via une requête POST.</li> <li>6. Le contrôleur vérifie la validité de chaque champ requis (nom, type d'assurance, date, etc.).</li> <li>7. Si tous les champs sont valides, la souscription est enregistrée dans la base de données.</li> <li>8. Le contrôleur renvoie un message de succès.</li> <li>9. L'interface affiche la confirmation de la souscription à l'utilisateur.</li> </ol>
<b>Enchaînements alternatifs</b>	<ul style="list-style-type: none"> <li>— Si un ou plusieurs champs sont invalides ou manquants, le contrôleur renvoie un message d'erreur.</li> <li>— L'interface affiche ce message à l'utilisateur, qui doit corriger les informations.</li> </ul>
<b>Postcondition</b>	La souscription à l'assurance est enregistrée dans la base de données et l'utilisateur reçoit une confirmation.

TABLE 3.19 – Description textuelle du cas d'utilisation Souscrire une assurance

### 3.4.4 Diagramme de séquence de Cas d'utilisation souscrire une assurance

La figure 3.49 montre le diagramme de séquence de cas d'utilisation « Souscrire une assurance ».

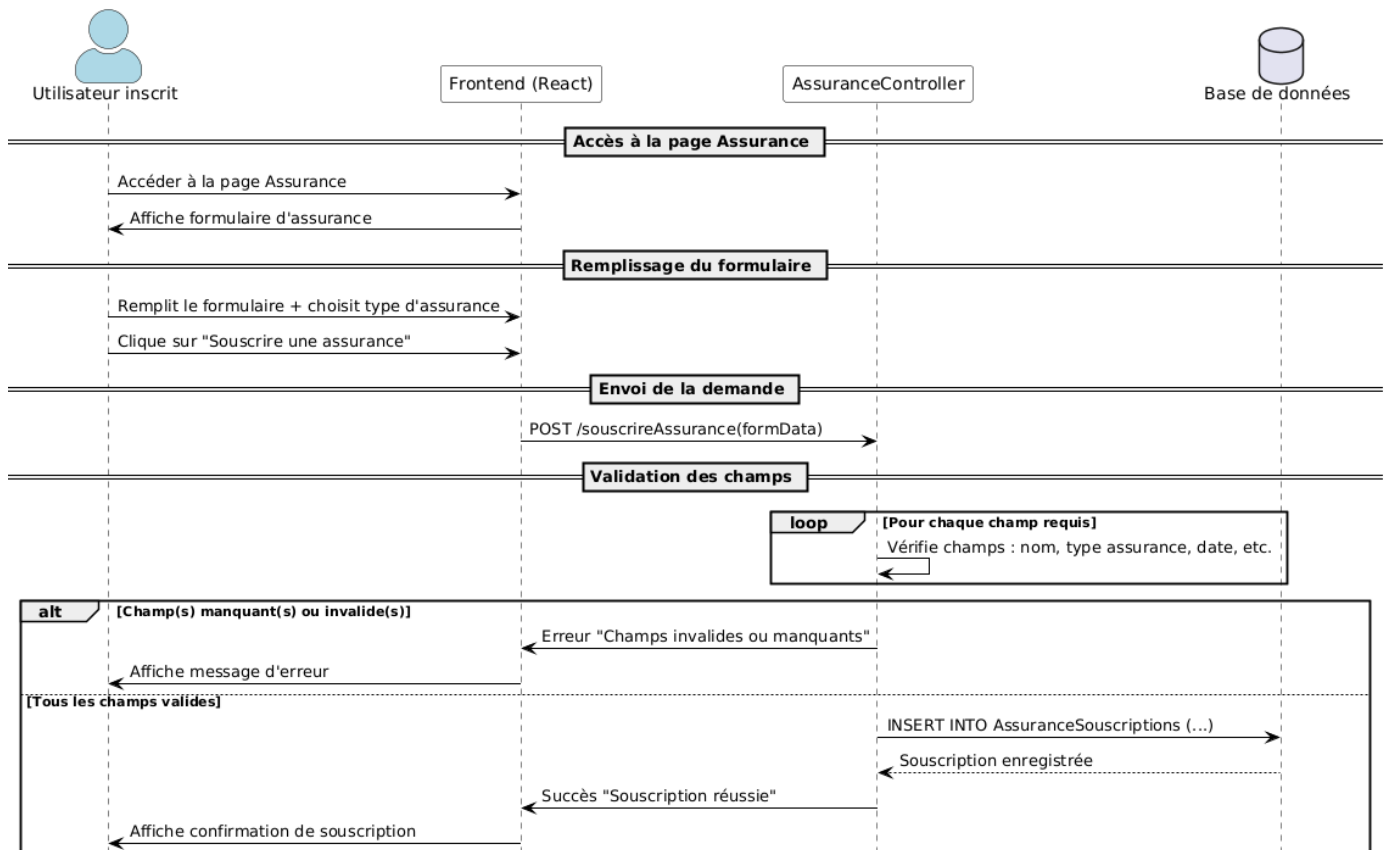


FIGURE 3.49 – Diagramme de séquence - Souscrire une assurance

### 3.4.5 Cas d'utilisation Consulter la liste des contrats d'assurance

#### 3.4.5.1 Description textuelle de cas d'utilisation Consulter la liste des contrats d'assurance

Le tableau 3.20 montre la description textuelle de cas d'utilisation « Consulter la liste des contrats d'assurance ».

<b>Cas d'Utilisation</b>	Consulter la liste des contrats d'assurances
<b>But</b>	Permettre à un agent ou un administrateur de visualiser tous les contrats d'assurances enregistrés.
<b>Acteur principal</b>	Agent / Administrateur
<b>Acteurs secondaires</b>	Aucun
<b>Préconditions</b>	L'acteur est authentifié et dispose des droits nécessaires pour accéder à la gestion des assurances.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'acteur accède à son tableau de bord après authentification.</li> <li>2. Il sélectionne la section "Assurances" dans le menu.</li> <li>3. Il clique sur "Liste des assurances".</li> <li>4. L'interface envoie une requête GET au contrôleur pour récupérer la liste.</li> <li>5. Le contrôleur interroge la base de données pour obtenir tous les contrats.</li> <li>6. Si la requête réussit, la liste est retournée et affichée à l'acteur.</li> </ol>
<b>Enchaînements alternatifs</b>	<ul style="list-style-type: none"> <li>— Si une erreur survient lors de la récupération (ex : problème de connexion à la base), un message d'erreur est retourné.</li> <li>— L'interface affiche alors ce message à l'acteur.</li> </ul>
<b>Postcondition</b>	La liste des contrats d'assurances est affichée, ou un message d'erreur est présenté en cas de problème technique.

TABLE 3.20 – Description textuelle du cas d'utilisation Consulter la liste des contrats d'assurance

### 3.4.5.2 Diagramme de séquence de Cas d'utilisation Consulter la liste des contrats d'assurance

La figure 3.50 montre le diagramme de séquence de cas d'utilisation « Consulter la liste des contrats d'assurance ».

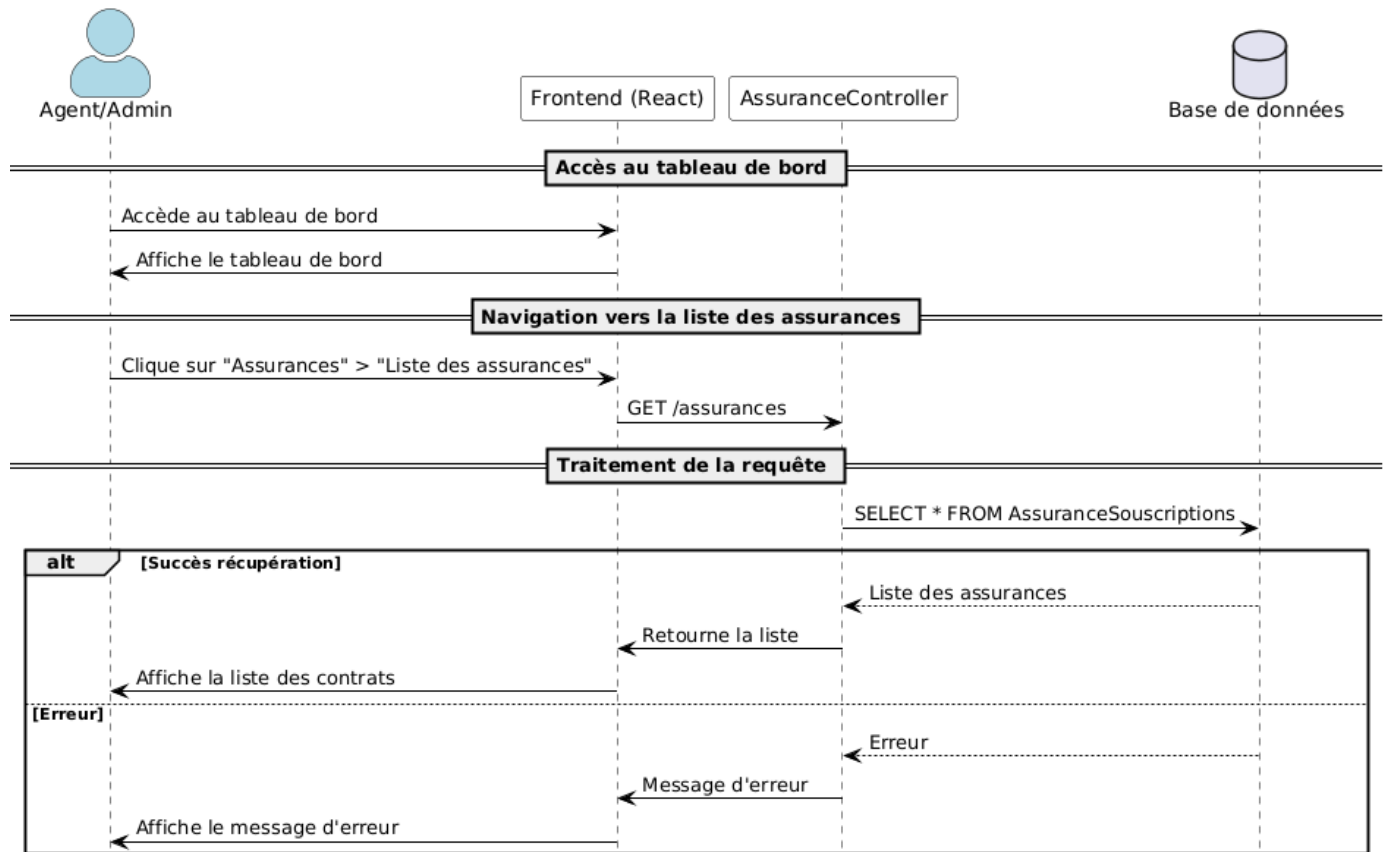


FIGURE 3.50 – Diagramme de séquence - Consulter la liste des contrats d'assurance

### 3.4.6 Cas d'utilisation Soumettre une demande de Visa

#### 3.4.6.1 Description textuelle de cas d'utilisation Soumettre une demande de Visa

Le tableau 3.21 montre la description textuelle de cas d'utilisation « Soumettre une demande de Visa ».

<b>Cas d'utilisation</b>	Soumettre une demande de Visa
<b>But</b>	Permettre à un utilisateur inscrit de remplir un formulaire de demande de visa avec justificatifs, pour lui-même et d'autres participants éventuels.
<b>Acteur principal</b>	Utilisateur inscrit
<b>Acteurs secondaires</b>	Interface (formulaire), VisaController, VisaModel, ParticipantModel, JustificatifModel
<b>Préconditions</b>	<ul style="list-style-type: none"> <li>— L'utilisateur doit être connecté.</li> <li>— Les données du formulaire doivent être complètes et valides.</li> </ul>
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur remplit un formulaire Visa avec les données personnelles et les justificatifs.</li> <li>2. Il soumet le formulaire via une requête POST vers <code>/visa</code>.</li> <li>3. Le contrôleur vérifie la validité des données.</li> <li>4. Une entrée Visa est créée avec les données générales.</li> <li>5. Pour chaque participant, une entrée est créée et liée au visa.</li> <li>6. Pour chaque justificatif, une entrée est créée et liée à chaque participant.</li> <li>7. Une réponse HTTP 201 est retournée, indiquant que le visa a été créé avec succès.</li> </ol>
<b>Enchaînement alternatif</b>	<p><b>[Données invalides] :</b></p> <ul style="list-style-type: none"> <li>— Le système retourne une erreur HTTP 400 avec un message décrivant les champs invalides ou manquants.</li> </ul>
<b>Postcondition</b>	La demande de visa est enregistrée, les participants et leurs justificatifs sont liés correctement, et une confirmation de création est renvoyée.

TABLE 3.21 – Description textuelle du cas d'utilisation Soumettre une demande de Visa

### 3.4.6.2 Diagramme de séquence de Cas d'utilisation prendre un rendez-vous visa

La figure 3.51 montre le diagramme de séquence de cas d'utilisation « Prendre un rendez-vous Visa ».

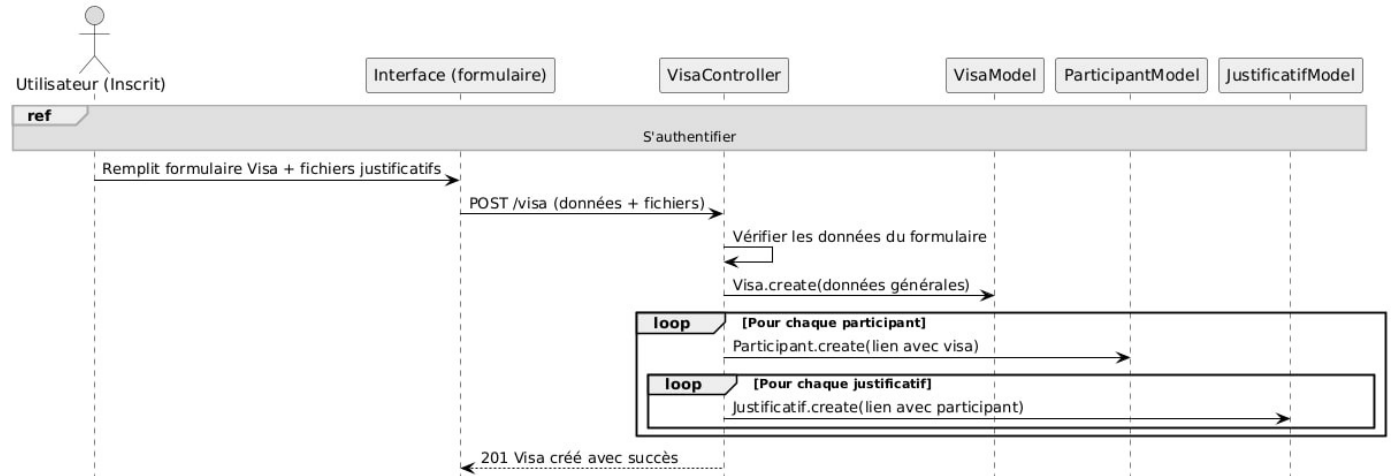


FIGURE 3.51 – Diagramme de séquence - Soumettre une demande de Visa

### 3.4.7 Cas d'utilisation Consulter la liste des visas

#### 3.4.7.1 Description textuelle de cas d'utilisation

Le tableau 3.22 montre la description textuelle de cas d'utilisation « Consulter la liste des visas ».

<b>Cas d'utilisation</b>	Consulter la liste des demandes de visa et afficher les détails des participants avec leurs justificatifs
<b>But</b>	Permettre à un administrateur ou agent de visualiser toutes les demandes de visa avec les utilisateurs associés, consulter les participants liés à une demande donnée, et voir les justificatifs fournis.
<b>Acteur principal</b>	Administrateur / Agent
<b>Acteurs secondaires</b>	/
<b>Préconditions</b>	<ul style="list-style-type: none"> <li>— L'administrateur est authentifié.</li> <li>— Des demandes de visa existent dans la base de données.</li> </ul>
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'administrateur clique sur "Liste des visas".</li> <li>2. Le système envoie une requête <code>GET /visas?search=&amp;pays=...&amp;typeVisa=...</code></li> <li>3. Le <code>VisaController</code> utilise <code>Visa.findAndCountAll()</code> avec filtres et jointures vers l'utilisateur inscrit.</li> <li>4. Le résultat paginé est renvoyé à l'interface administrateur.</li> <li>5. L'administrateur clique sur "Détails" d'un visa.</li> <li>6. Une requête <code>GET /visas/:id/participants</code> est envoyée.</li> <li>7. Le <code>VisaController</code> appelle <code>Participant.findAll({ visaId })</code>.</li> <li>8. La liste des participants est retournée.</li> <li>9. Pour chaque participant, une requête <code>GET /participants/:id/justificatifs</code> est faite.</li> <li>10. Le contrôleur appelle <code>Justificatif.findAll({ participantId })</code>.</li> <li>11. La liste des justificatifs est retournée et affichée.</li> </ol>
<b>Enchaînements alternatifs</b>	<ul style="list-style-type: none"> <li>— Aucun participant trouvé pour un visa : message "Aucun participant associé".</li> <li>— Aucun justificatif pour un participant : message "Aucun justificatif transmis".</li> </ul>
<b>Postcondition</b>	L'administrateur peut consulter tous les visas, les participants associés et leurs justificatifs, pour traiter ou valider les demandes.

TABLE 3.22 – Description textuelle du cas d'utilisation Consulter la liste des visas

### 3.4.7.2 Diagramme de séquence de Cas d'utilisation consulter la liste des visas

La figure 3.52 montre le diagramme de séquence de cas d'utilisation « Consulter la liste des visas ».

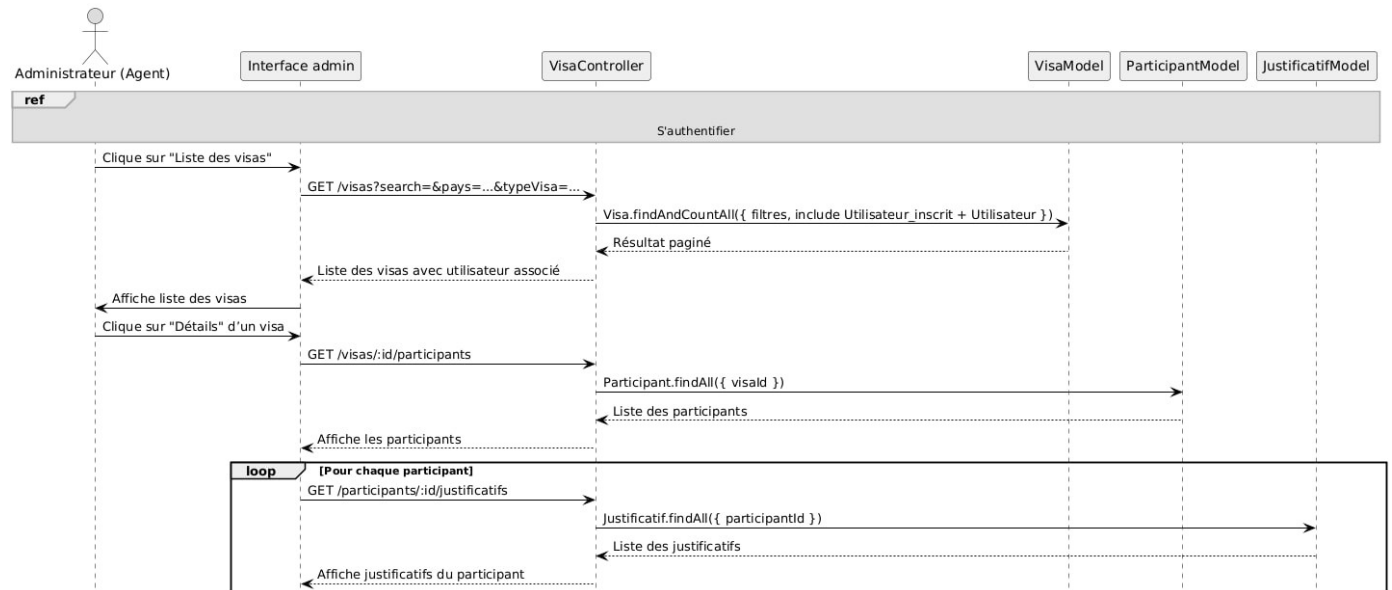


FIGURE 3.52 – Diagramme de séquence - Consulter la liste des visas

### 3.4.8 Diagramme de Classe 3ème sprint

La figure 3.53 montre le diagramme de classe de 3ème sprint.

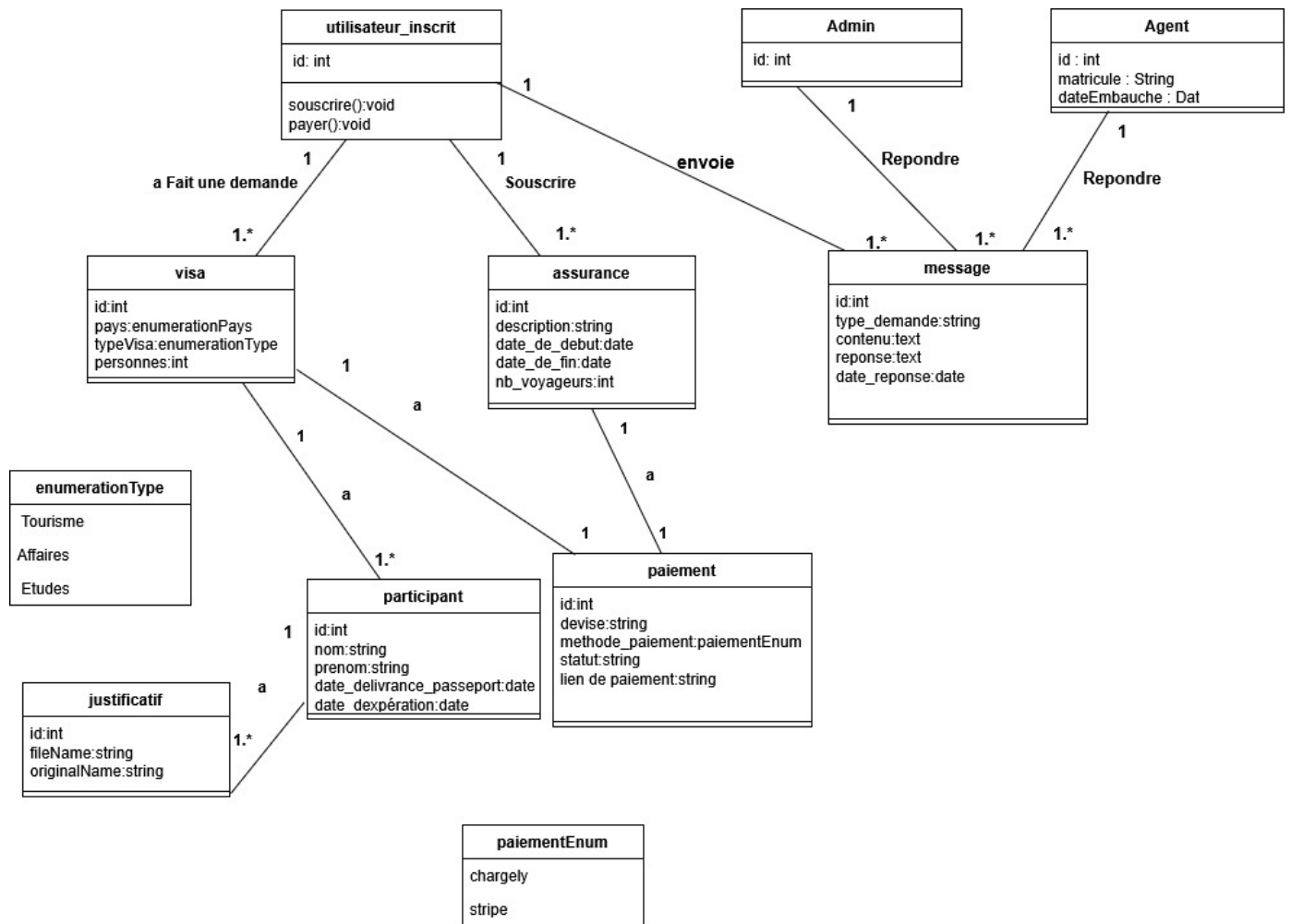


FIGURE 3.53 – Diagramme de classe - 3ème sprint

### 3.4.9 Modèle relationnel de 3ème sprint

Voici le modèle relationnel correspondant au diagramme de classe de 3ème sprint.

**Utilisateur\_inscrit**(id,)

**Assurance**(id, description, prix, dateDebut, dateFin, nombreVoyageurs, passeport, type, statutValidation, #id\_utilisateur\_inscrit)

**Visa**(id, pays, typeVisa, personnes, nationalite, dateArrivee, accepteConditions, total, statutValidation, #utilisateurInscritId)

**Participant**(id, nom, prenom, date\_delivrance\_passeport, date\_expiration, #id\_visa)

**Justificatif**(id, fileName, originalName, #id\_participant)

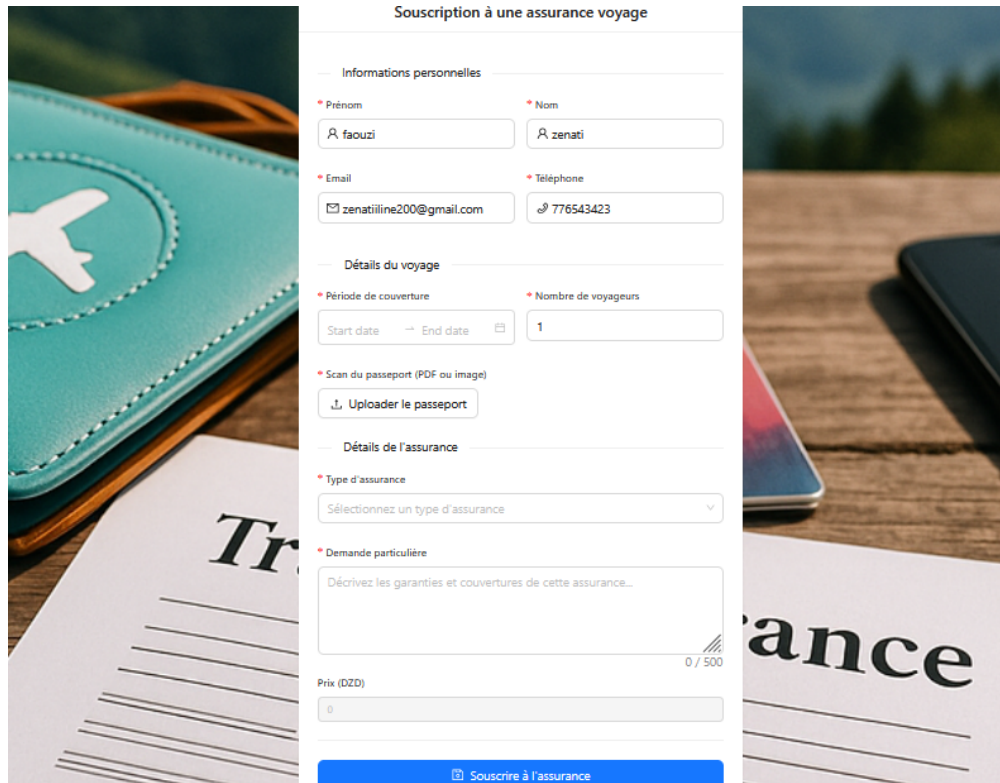
**Message**(id, typeDemande, contenu, reponse, dateReponse, #utilisateurInscritId, #id\_agent, #id\_admin)

**Paiement**(id, devise, methode\_paiement, statut, lien\_de\_paiement, #id\_assurance, #id\_visa)

### 3.4.10 Interfaces du troisième sprint

#### 3.4.10.1 Interface Souscrire une assurance

L'utilisateur inscrit peut souscrire à une assurance voyage proposée par l'agence via un formulaire simple.



The image shows a mobile application interface for subscribing to travel insurance. The form is titled "Souscription à une assurance voyage" and is divided into three main sections: "Informations personnelles", "Détails du voyage", and "Détails de l'assurance".

- Informations personnelles:** Includes fields for "Prénom" (faouzi), "Nom" (zenati), "Email" (zenatiilene200@gmail.com), and "Téléphone" (776543423).
- Détails du voyage:** Includes a "Période de couverture" field with "Start date" and "End date" pickers, and a "Nombre de voyageurs" field set to "1". There is also an "Upload le passeport" button.
- Détails de l'assurance:** Includes a "Type d'assurance" dropdown menu, a "Demande particulière" text area, and a "Prix (DZD)" field.

At the bottom of the form is a blue button labeled "Souscrire à l'assurance". The background of the screenshot shows a teal passport cover with a white airplane icon and a document with the word "assurance" visible.

FIGURE 3.54 – Interface de souscription à une assurance

#### 3.4.10.2 Interface Liste des contrats d'assurances

L'agent/admin peut consulter l'ensemble des contrats d'assurance.

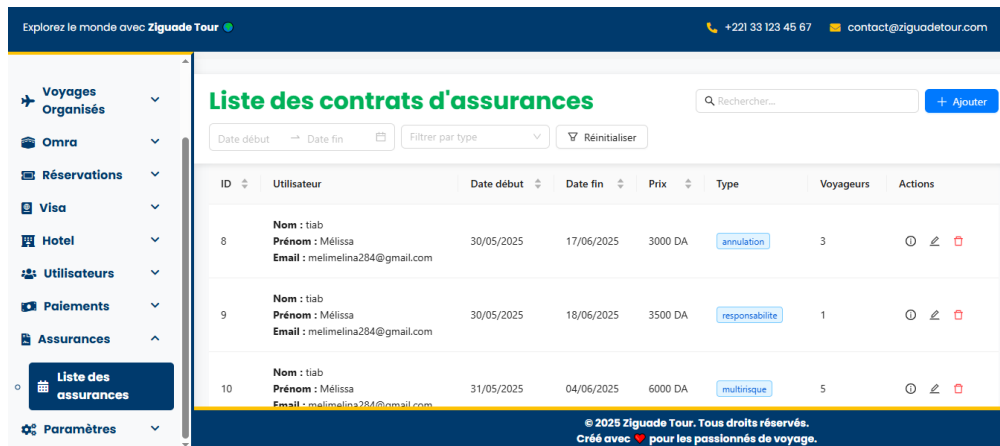


FIGURE 3.55 – Interface de consultation des assurances

### 3.4.10.3 Interface Demande de visa

L'utilisateur peut soumettre une demande de visa en ligne en remplissant un formulaire contenant ses informations personnelles, le pays de destination, les justificatifs à joindre, etc.

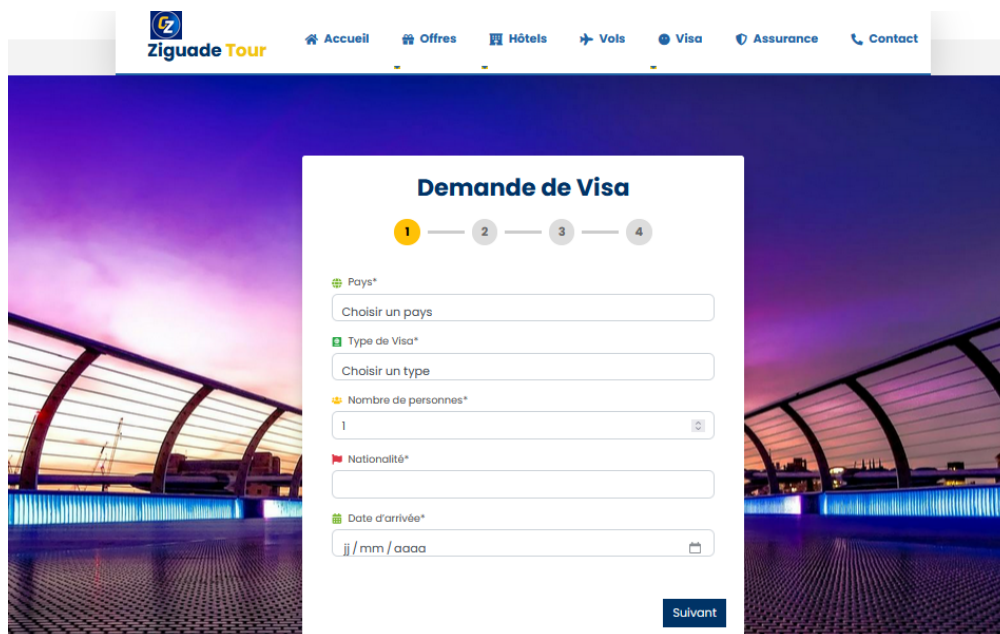


FIGURE 3.56 – Interface de soumission d'une demande de visa

### 3.4.10.4 Interface Liste des visas

Cette interface permet à l'agent/admin de visualiser la liste des visas

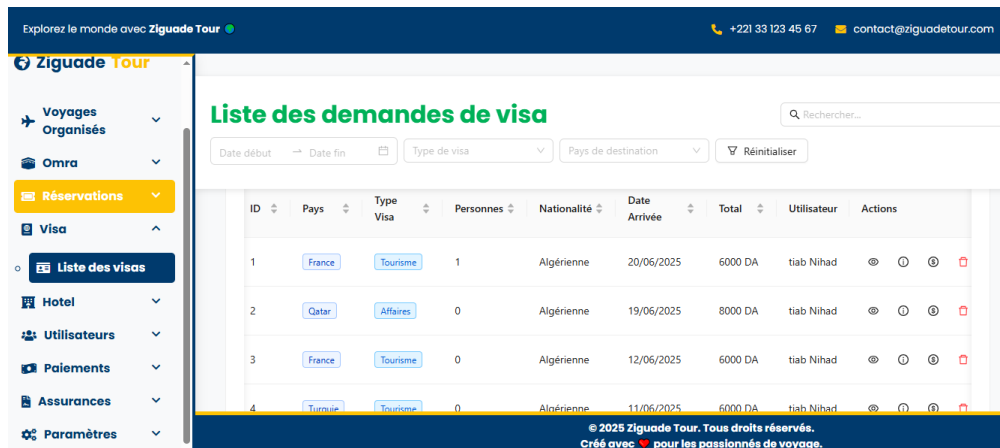


FIGURE 3.57 – Interface de consultation des offres de visa

### 3.4.10.5 Interface de formulaire de contact

Cette interface permet à l'utilisateur\_inscrit d'envoyer un message à l'agent/admin.

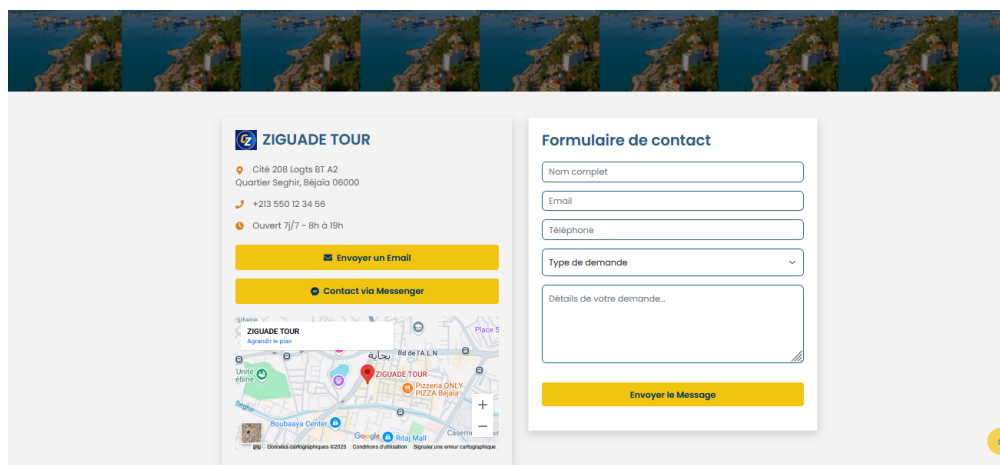


FIGURE 3.58 – Interface de formulaire de contact

### 3.4.10.6 Interface de répondre a un message

Cette interface permet à l'utilisateur\_inscrit d'envoyer un message à l'agent ou à l'administrateur.

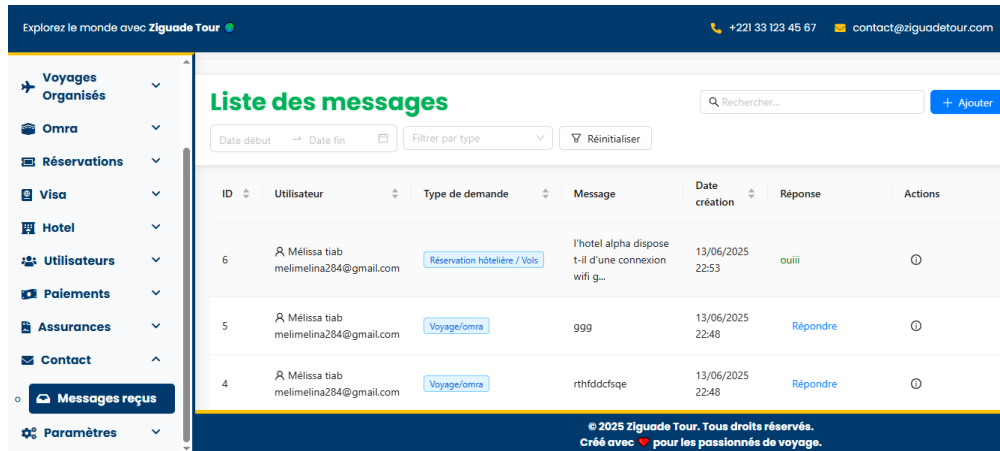


FIGURE 3.59 – Interface de répondre a un message

## 3.5 Conclusion

Ce chapitre a constitué une étape essentielle dans le développement de l'application en présentant nos choix techniques et les principales fonctionnalités mises en place. L'utilisation de Node.js, React, ainsi que l'intégration de services comme Chargily ePay, Amadeus et l'API Graph de Facebook, nous a permis de concevoir une application complète et connectée. La séparation claire entre backoffice et frontoffice, associée à des interfaces ergonomiques, garantit une expérience utilisateur fluide et intuitive.

# Chapitre 4: Intégration des APIs externes : Réseaux sociaux et Paiement

## 4.1 Introduction

Ce chapitre présente l'intégration des API essentielles utilisées dans notre application web. Il couvre d'une part les API de Facebook et Instagram (Meta Graph API), permettant l'automatisation de la publication, la récupération et la modération de contenus sociaux; et d'autre part les solutions de paiement Stripe et Chargily, utilisées pour sécuriser les transactions en ligne. Nous détaillons les prérequis, les appels API, les contraintes techniques, ainsi que les outils de test utilisés dans chaque cas.

## 4.2 Intégration des API de Facebook et Instagram

### 4.2.1 Présentation de Meta Graph API

Dans le cadre de notre application, nous avons intégré deux APIs externes fournies par la plateforme Meta for Developers : la Facebook Graph API et la Instagram Graph API. Ces deux interfaces permettent d'automatiser la publication de contenu, de synchroniser les interactions des utilisateurs (commentaires, likes), et d'assurer une meilleure visibilité des offres touristiques sur les réseaux sociaux.

#### 4.2.1.1 Facebook Graph API

L'API Graph est le seul moyen d'insérer et de récupérer des données dans la plateforme Facebook. Il s'agit d'une API basée sur le protocole HTTP qui permet aux apps d'avoir recours à la programmation pour interroger des données, publier de nouvelles actualités, gérer des publicités, Lire les interactions (commentaires, réactions, partages), Supprimer ou masquer du contenu.[21] Basée sur le protocole HTTP, l'API accepte des requêtes simples au format **URL** (GET, POST, DELETE, etc.) et retourne des réponses au format **JSON**. C'est actuellement le moyen le plus complet, flexible et puissant pour accéder aux fonctionnalités offertes par les plateformes Meta. Dans notre projet, elle a été utilisée pour :

- Publication automatique des offres de voyages organisés et des Omra sur la page Facebook de l'agence.
- Synchronisation et récupération des commentaires ainsi que des réactions (likes) associés à chaque publication.
- Suppression des commentaires indésirables directement depuis l'interface de gestion.
- Accès direct, depuis une publication sur Facebook, à la version correspondante sur le site web.

- Navigation inversée possible : depuis un commentaire affiché sur le site, accès à la publication correspondante sur Facebook.

#### 4.2.1.2 Instagram Graph API

L'Instagram Graph API permet aux applications d'interagir avec des comptes professionnels (Business) ou créateurs (Creator) sur Instagram, à condition que ces comptes soient reliés à une page Facebook. Il s'agit d'une API RESTful, tout comme la Facebook Graph API, reposant sur le protocole HTTP et acceptant des requêtes simples au format URL (GET, POST, DELETE, etc.), avec des réponses au format JSON.

Cette API permet aux développeurs d'insérer et de récupérer des contenus publiés sur un compte professionnel, de gérer et répondre aux commentaires, de rechercher des publications contenant des hashtags spécifiques, ainsi que d'accéder à des métadonnées et à des statistiques de base sur d'autres comptes professionnels ou créateurs. Son intégration est essentielle pour automatiser la gestion des interactions sociales au sein d'une application tierce [22].

Dans notre projet, l'Instagram Graph API a été utilisée pour les mêmes fonctionnalités que celles mises en œuvre avec la Facebook Graph API, notamment la publication automatique, la gestion des commentaires et la synchronisation avec le site web.

### 4.2.2 Conditions d'utilisation des comptes pour accéder aux APIs Meta

L'accès aux APIs de la plateforme Meta (Facebook et Instagram) repose sur des exigences différentes selon le réseau ciblé.

Pour Facebook, il suffit qu'un utilisateur dispose d'un compte personnel et crée ou administre une Page Facebook. Une fois cette page créée, le compte peut être authentifié via le protocole OAuth, ce qui permet d'obtenir un jeton d'accès valide pour utiliser l'API Graph. Aucune conversion du compte personnel n'est requise pour exploiter les principales fonctionnalités de l'API telles que la publication, la récupération de commentaires ou la gestion des interactions sur la page [23].

En revanche, pour Instagram, Meta impose des conditions plus strictes. L'accès à l'Instagram Graph API est réservé exclusivement aux comptes professionnels ou créateurs (Business ou Creator). De plus, ce compte Instagram doit être obligatoirement associé à une Page Facebook via le centre professionnel (Meta Business Suite). Cette double condition est indispensable pour que les jetons d'accès soient valides et pour que l'API puisse interagir avec le compte Instagram [22].

### 4.2.3 Authentification et génération des jetons

L'utilisation de la Graph API nécessite un jeton d'accès (access token), qui permet d'authentifier les requêtes. Il en existe trois principaux types [24] :

**Jeton utilisateur** : temporaire, lié à un compte Facebook personnel, utilisé pour authentifier l'utilisateur via OAuth.

**Jeton de page** : dérivé du jeton utilisateur, permet d'interagir avec une Page Facebook, indispensable pour publier du contenu ou lire/modérer les commentaires.

**Jeton d'application** : utilisé dans les cas où l'accès à des données publiques ou des tâches administratives ne nécessite pas d'utilisateur.

Dans notre projet, nous utilisons principalement un jeton de page à long terme, obtenu via la console Meta for Developers, pour automatiser la publication de contenus sur la Page Facebook professionnelle de l'agence, ainsi que sur le compte Instagram professionnel qui y est associé, le tout à partir d'un backend Node.js sécurisé.

## 4.2.4 Publication et gestion des commentaires

Cette section décrit les mécanismes d'interaction mis en place pour publier des contenus et gérer les commentaires sur les deux plateformes Meta : Facebook et Instagram, en s'appuyant sur les APIs respectives.

### 4.2.4.1 Sur Facebook

Pour publier sur une Page Facebook, la Graph API propose plusieurs points d'accès selon le type de contenu[25] :

Pour une publication texte ou lien :

```
POST /{page-id}/feed
```

Pour une image :

```
POST /{page-id}/photos
```

Exemple de requête :

```
POST https://graph.facebook.com/v19.0/{page-id}/photos
Authorization: Bearer {access_token}
Content-Type: application/json

{
  "url": "https://site.com/image.jpg",
  "caption": "D couvrez notre nouveau voyage !"
}
```

Une fois la publication en ligne, les commentaires sont récupérés via[26] :

```
GET /{post-id}/comments?access_token={token}
```

Et peuvent être supprimés :

```
DELETE /{comment-id}?access_token={token}
```

### 4.2.4.2 Sur Instagram

La publication via l'Instagram Graph API s'effectue en deux étapes [27] :

1. Création du conteneur média :

```
POST /{ig-user-id}/media
```

2. Publication du média créé :

```
POST /{ig-user-id}/media_publish
```

Instagram ne permet pas d'ajouter de lien cliquable dans la légende. Le lien peut uniquement figurer dans la bio du profil ou être utilisé dans une campagne publicitaire.

Récupération des commentaires [28] :

```
GET /{ig-media-id}/comments?access_token={token}
```

Suppression d'un commentaire :

```
DELETE /{comment-id}?access_token={token}
```

## 4.2.5 Limitations d'utilisation des APIs Meta (Quotas)

Les plateformes Meta imposent des quotas afin de réguler l'usage des APIs et d'éviter les abus. Ces limitations varient selon le type de compte, l'application utilisée et la plateforme ciblée (Facebook ou Instagram).

### Facebook Graph API

Pour la Facebook Graph API, les limitations sont calculées dynamiquement en fonction du nombre d'utilisateurs actifs de l'application. En règle générale, une application peut effectuer environ **200 requêtes par heure et par utilisateur actif**. Cela signifie que si une application compte 100 utilisateurs actifs, elle peut théoriquement envoyer jusqu'à 20 000 requêtes par heure.

L'utilisation est surveillée via des en-têtes HTTP spécifiques retournés dans chaque réponse : `X-App-Usage`, `X-Page-Usage`, etc., permettant aux développeurs de suivre en temps réel la consommation de leur quota [29].

Dans notre application, nous avons mis en place des mécanismes pour limiter le nombre d'appels à l'API Facebook, notamment pour :

### Instagram Graph API

L'Instagram Graph API applique des limitations plus strictes. En plus des 200 appels/heure par utilisateur actif (comme Facebook), elle impose une **limite de 100 publications (posts, carrousels, vidéos)** maximum toutes les 24 heures pour chaque compte professionnel (Business ou Creator). Certaines actions comme la gestion des commentaires sont limitées à **60 écritures par heure et par utilisateur** [30].

Dans notre projet, nous avons respecté ces contraintes en limitant :

- le nombre de publications d'Omra et de voyages par jour sur Instagram,
- le déclenchement des appels de récupération de commentaires et likes à des intervalles contrôlés,
- et la suppression de commentaires uniquement sur demande explicite de l'administrateur.

## 4.2.6 Gestion des erreurs courantes avec l'API Graph

lors de l'intégration de l'API Graph de Meta, il est impératif de prévoir un traitement robuste des erreurs retournées par le serveur. Chaque réponse d'erreur comprend un code d'erreur, un message explicatif, parfois un type d'erreur, ainsi que des conseils pour la résolution. Voici les cas les plus fréquents [31]. :

**Erreur 190** – Invalid OAuth 2.0 Access Token : le jeton d'accès est invalide, expiré ou mal formé. Conserver ou renouveler le token.

**Erreur 10** – Permission denied : permissions manquantes pour l'action demandée. Vérifier que l'utilisateur a accordé les scopes requis et que l'application est approuvée.

**Erreur 200** – Permissions error : tentative d'accès à une ressource sans les droits adéquats (ex. publier sur une page sans être admin).

**Erreur 102** – Session Expired : le jeton a expiré. Il faut le renouveler via le mécanisme de tokens à long terme.

**Erreur 613** – Calls to this API have exceeded the rate limit : le quota de requêtes a été dépassé. Il est conseillé d'utiliser un mécanisme de throttling et de surveiller les quotas via le dashboard développeur.

**Erreur 100** – Invalid parameter : paramètres requis manquants ou mal formatés (ex. ID incorrect). Consulter la documentation des endpoints.

**Erreur 368** – Temporarily blocked : action bloquée temporairement en raison d'un comportement suspect ou abusif (ex. publications trop fréquentes). Il faut temporiser ou contacter le support Meta.

**Erreur 4** – Application request limit reached : surcharge du serveur ou restrictions globales. Cette erreur est souvent passagère.

Dans notre intégration Node.js, nous détectons systématiquement les erreurs HTTP (codes 400 à 599) et les journalisons côté serveur. Les erreurs critiques (comme 190 ou 10) sont remontées au front-end pour informer l'utilisateur. Voici un exemple de réponse JSON typique analysée :

```
{
  "error": {
    "message": "Message describing the error",
    "type": "OAuthException",
    "code": 190,
    "error_subcode": 460,
    "fbtrace_id": "EJplcsCHuLu"
  }
}
```

Listing 4.1 – Exemple de message d'erreur retourné par l'API Graph

## 4.2.7 Utilisation de Ngrok pour les tests en local

Dans notre projet, nous avons eu besoin d'assurer la communication entre les publications diffusées sur **Facebook** et **Instagram**, et les pages correspondantes sur notre site web. Pour permettre cette interaction — notamment l'accès depuis une publication sur un réseau social vers sa version hébergée localement sur notre serveur Node.js — il était indispensable de rendre nos routes accessibles publiquement.

Cependant, tant que notre application n'est pas déployée en ligne, les URL locales (`http://localhost`) ne sont pas accessibles depuis les serveurs de Meta. Pour contourner cette limitation, nous avons utilisé l'outil **Ngrok**, qui génère une URL HTTPS publique et temporaire pointant vers notre backend local. Exemple d'URL générée :

```
https://abcdef1234.ngrok.io/publication/64e7b6f83fb51e0015a1f53c
```

Listing 4.2 – Exemple d'URL Ngrok utilisée pour les tests de redirection

Cette URL a été utilisée dans notre projet pour permettre aux utilisateurs de naviguer depuis une publication Facebook ou Instagram vers la page détaillée correspondante sur le site (toujours en local), et inversement.

Ngrok s'est donc révélé essentiel pour tester cette interaction en environnement de développement, en attendant le déploiement de l'application sur un serveur de production sécurisé. [32]

## 4.3 Intégration des API de paiement Stripe et Chargily

### 4.3.1 Présentation des solutions

#### 4.3.1.1 Stripe

Stripe est une plateforme de paiement en ligne à portée internationale destinée aux entreprises internationales. Elle permet d'accepter les principales cartes bancaires (Visa, Mastercard, Carte Bleue, etc.) dans la plupart des pays du monde (195 pays) et de traiter les paiements dans plus de 135 devises différentes. Grâce à son infrastructure mondiale (Stripe est disponible dans 46 pays et s'étend régulièrement), Stripe s'adapte à de nombreux cas d'usage : boutiques en ligne multidevise, marketplaces internationales, etc. [33]

#### 4.3.1.2 Chargily Pay

Chargily Pay est une passerelle de paiement algérienne dédiée au marché local. Elle cible les commerçants opérant en Algérie et traitant leurs paiements en dinars algériens (DZD), en proposant notamment le paiement par les cartes bancaires nationales CIB et la carte magnétique EDAHABIA (Algérie Poste). L'API de Chargily Pay permet de créer des sessions de paiement (checkout) en dinars algériens (DZD). Le site officiel de Chargily indique qu'il s'agit d'une solution sécurisée, avec une API gratuite et des modules d'intégration (plugins) disponibles pour faciliter la mise en place. Ainsi, Chargily convient aux boutiques ou services ciblant une clientèle algérienne. [34]

### 4.3.2 Inscription et génération des clés API

#### 4.3.2.1 Stripe

Pour utiliser Stripe, il faut d'abord créer un compte sur `stripe.com`. Lors de l'inscription, on fournira les informations de l'entreprise (forme juridique, secteur, coordonnées) et

un compte bancaire professionnel pour les virements de fonds. Après vérification (qui est généralement rapide), le compte Stripe sera activé. Ensuite, dans le Dashboard Stripe (tableau de bord développeur), on se rend dans la section Développeurs > Clés API. Stripe y fournit deux paires de clés : une clé publiable (pk\_...) à utiliser côté client (pour initialiser Stripe.js) et une clé secrète (sk\_...) à utiliser côté serveur. Par défaut, chaque environnement (test et production) dispose d'une clé secrète et d'une clé publiable. Ces clés se trouvent dans l'onglet Clés API du Dashboard. Il suffit de copier les valeurs et de les stocker de manière sécurisée (par exemple, dans des variables d'environnement) pour authentifier toutes les requêtes ultérieures vers l'API Stripe.

### 4.3.2.2 Chargily

De même, pour utiliser Chargily Pay, nous avons créé un compte sur la plateforme. Le processus d'inscription s'est révélé simple et rapide : il suffit de se rendre sur la page d'inscription [pay.chargily.com/dashboard/register](https://pay.chargily.com/dashboard/register), de saisir les informations demandées et de valider l'adresse e-mail via le lien reçu. L'inscription peut également se faire rapidement en se connectant avec un compte Google ou GitHub sur la page de connexion. Une fois le compte Chargily Pay créé, l'utilisateur est automatiquement placé en mode Test. Dans le Dashboard Chargily Pay, on peut alors créer une ou plusieurs Applications (un concept qui permet de gérer plusieurs projets ou sites sous un même compte). Chaque application dispose de ses propres clés API. Pour obtenir les clés, on se rend sur la page Developers Corner du Dashboard Chargily (via le lien « Dashboard »), où sont affichées les clés publique et secrète pour le mode actuellement sélectionné (Test ou Live). Chargily fournit également deux URL de base correspondant aux modes Test et Live (par exemple <https://pay.chargily.net/test/api/v2> pour Test). Il faut noter qu'après la création du compte, on travaille d'abord en mode Test. Pour passer en mode Live, il faut d'abord vérifier son compte Chargily (soumission des pièces justificatives) ; une fois l'intégration testée, on bascule simplement sur l'URL et les clés du mode Live dans le Dashboard.

### 4.3.3 Intégration technique

Notre intégration suit les bonnes pratiques : le backend de notre application interagit directement avec les APIs, en conservant les clés secrètes sur le serveur.

- **Stripe** : Nous avons utilisé la bibliothèque officielle Stripe (disponible en Node.js, PHP, etc.) et configuré la clé secrète appropriée. Pour lancer un paiement, notre serveur créait un objet de paiement (par exemple un `PaymentIntent` ou une `Checkout Session`) via l'API Stripe. Il transmettait ensuite au front-end l'URL ou les données nécessaires pour afficher le formulaire de paiement (Stripe.js ou Stripe Checkout). Les callbacks (webhooks) ont été configurés afin de recevoir automatiquement les notifications de succès ou d'échec de paiement.
- **Chargily** : Nous avons intégré l'API REST Chargily V2 en appelant l'endpoint `POST /checkouts`. Concrètement, le serveur envoyait une requête HTTPS à [https://pay.chargily.net/\[mode\]/](https://pay.chargily.net/[mode]/) avec un en-tête `Authorization : Bearer <le_token>` et un corps JSON décrivant la transaction (montant, devise, pages de succès/échec, etc.). Par exemple, en mode test :

```
POST https://pay.chargily.net/test/api/v2/checkouts
{ "amount": 2500, "currency": "dzd",
  "success_url": "https://notre-site.com/success", ... }
```

La réponse renvoie un objet JSON incluant notamment un champ `checkout_url` (une URL Chargily où l'utilisateur sera redirigé pour finaliser le paiement). Nous redirigeons ensuite l'utilisateur vers cette URL pour qu'il saisisse ses informations de paiement. L'API Chargily propose également des champs pour personnaliser la demande de paiement. Dans notre cas, nous nous sommes limités aux champs principaux requis par l'API : le montant de la transaction (`amount`) et les informations du client. L'intégration s'effectue en REST/JSON sans besoin de bibliothèque dédiée, ce qui reste accessible et clair du point de vue développeur.

Dans les deux cas, nous avons pris soin de tester les appels API avec les clés de test, de vérifier le format des réponses, et d'implémenter la gestion des webhooks pour enregistrer en base de données l'état final de chaque transaction.

#### 4.3.4 Tests

Nous avons procédé à de nombreux tests

- **Stripe (mode test)** : en mode test, Stripe nous permet d'effectuer des paiements simulés sans transfert réel d'argent. Nous utilisons les cartes virtuelles de test fournies par Stripe pour vérifier tous les scénarios (paiement autorisé, refusé, etc.). Ce mode est entièrement gratuit puisque les transactions ne sont pas réelles. Il garantit la fiabilité de l'intégration avant de passer en live.
- **Chargily (mode test)** : de même, Chargily propose un environnement test (basé sur l'URL `.../test/api/v2`) où les paiements sont simulés. Nous avons testé la création de checkout, la redirection vers une URL de paiement simulée (utilisée uniquement en mode test). Aucune somme n'est débitée en mode test. La documentation Chargily confirme que l'intégration et les tests sont totalement gratuits. En mode test, on reste en dessous des limites du plan gratuit (0% de frais), cela nous permet de tester tout le processus de paiement du début à la fin sans frais.

Dans les deux environnements de test, aucun frais n'est prélevé : Stripe ne facture rien en mode sandbox, et Chargily ne prélève pas de commission (0%) sur les transactions test ou sur le plan de base. Cela nous a permis de mener des tests rigoureux à moindre coût.

#### 4.3.5 Différences principales

Nous soulignons ci-dessous les points de comparaison clés entre les deux APIs [33] [34]

- **Coût des transactions** : Stripe ne prélève pas de frais pour l'utilisation de l'API en mode test, mais en production chaque paiement a un coût standard (environ 2,90% + 0,30 USD par transaction réussie). En revanche, Chargily propose un plan gratuit à 0% de commission. Selon la documentation Chargily, l'intégration et les paiements

restent complètement gratuits tant qu'on reste sur le plan de base. Autrement dit, nous n'avons aucune commission Chargily sur les transactions de base, ce qui est un avantage financier pour l'entreprise.

- **Couverture géographique et devises** : Stripe est disponible dans plus de 195 pays et supporte plus de 135 devises. Nous pouvons ainsi vendre à l'international et accepter différentes monnaies. Chargily, en revanche, est centré sur l'Algérie : sa devise supportée est uniquement le dinar algérien (DZD). Il s'agit d'une solution adaptée aux marchés locaux, mais limitée si l'on souhaite cibler d'autres pays.
- **Moyens de paiement supportés** : Stripe prend en charge une large gamme de cartes bancaires internationales (Visa, MasterCard, Carte Bleue, etc.), ainsi que divers wallets et méthodes locales (Apple Pay, Google Pay, Alipay, etc. via ses intégrations). Chargily prend pour l'instant en charge uniquement les cartes nationales algériennes EDAHA-BIA et CIB, ainsi qu'un code QR de l'application Chargily. Par exemple, en créant un checkout Chargily, on spécifie le champ "payment\_method" ("edahabia", "cib" ou "chargily\_app"). Le client peut ensuite choisir son option. Cette spécialisation signifie que Chargily est idéal pour un public algérien (paiement en mode local), tandis que Stripe s'adresse à un public plus large.
- **Langues et régionalisation** : L'interface de paiement Chargily peut s'afficher en arabe ou anglais, ce qui est utile pour des utilisateurs maghrébins. Stripe propose également des interfaces multilingues pour s'adapter aux différentes langues. Les deux solutions offrent une expérience adaptée au contexte linguistique et culturel de l'utilisateur.

### 4.3.6 Gestion des Webhooks

Les webhooks jouent un rôle crucial dans les intégrations de paiements : ils permettent à notre application backend de recevoir des notifications automatiques dès qu'un événement important survient côté prestataire (paiement réussi, échec, etc.).

### 4.3.7 Webhooks Stripe

Stripe utilise un système de webhooks riche et sécurisé. Lorsqu'un événement se produit (ex. : `payment_intent.succeeded`, `checkout.session.completed`, `payment_intent.payment_failed`), Stripe envoie une requête POST signée à une URL définie par le développeur (par exemple : `https://localhost:3000/webhook/stripe`).

Dans notre backend (Node.js), nous avons :

1. Défini une route spécifique `/webhook/stripe`.
2. Utilisé la bibliothèque Stripe pour valider la signature (`stripe.webhooks.constructEvent(...)`) en utilisant une clé secrète fournie dans le Dashboard (onglet Webhooks).
3. Traité les événements :
  - Si `payment_intent.succeeded` : nous marquons la réservation comme « terminée » et le paiement comme « payé »
  - Si `payment_intent.payment_failed` : nous notifions l'utilisateur de l'échec.

Cette approche garantit que les statuts sont mis à jour même si l'utilisateur interrompt sa navigation avant la confirmation finale du paiement.

### 4.3.8 Webhooks Chargily

Chargily offre également un support des webhooks de confirmation. Lorsqu'un client finalise son paiement, Chargily envoie une requête HTTP POST à une URL définie dans le champ "back\_url" lors de la création du checkout.

Voici notre configuration :

1. Lors de la création du checkout, nous avons défini le champ back\_url (ex. : https://localhost:3000.com/webhook/chargily).
2. Une fois le paiement confirmé par l'utilisateur, Chargily envoie à cette URL un objet JSON contenant les données de la transaction (ID, statut, montant, client, etc.).
3. Le backend valide ces données, vérifie leur cohérence avec la base de données, et met à jour le statut de la réservation en 'Terminée' et celui du paiement en 'Payé'.

## 4.4 Conclusion

L'intégration des API de Meta et des solutions de paiement a permis de renforcer les fonctionnalités de notre application tant sur le plan marketing que commercial. Chaque API a été testée dans un environnement sécurisé et respecte les exigences des plateformes partenaires. Cette mise en œuvre assure un fonctionnement fluide, sécurisé et conforme aux standards actuels, tout en restant évolutive pour de futures améliorations.

## Conclusion et Perspectives

La gestion d'une agence de voyage nécessite aujourd'hui une organisation efficace et l'utilisation de solutions informatiques adaptées. Ce mémoire a présenté les défis liés à la gestion des offres, la publication sur plusieurs plateformes, l'interaction avec les clients via les réseaux sociaux, ainsi que l'intégration sécurisée des paiements en ligne. Nous avons constaté que les systèmes traditionnels, souvent peu automatisés, limitent la réactivité et la qualité du service. Le développement d'une application web basée sur la méthode agile Scrum permet une meilleure organisation du travail, une adaptation rapide aux besoins, et une amélioration continue des processus. La conception d'une architecture technique solide, accompagnée d'une modélisation précise, a été essentielle pour répondre aux besoins fonctionnels et garantir une expérience utilisateur satisfaisante. L'intégration des réseaux sociaux et des moyens de paiement en ligne est devenue indispensable pour renforcer la visibilité, la confiance et la fidélité des clients.

En conclusion, la transformation digitale des agences de voyage est un facteur clé pour leur développement et leur compétitivité. Ce travail ouvre des perspectives pour intégrer à l'avenir des technologies innovantes, afin d'accompagner la croissance du secteur touristique.

Parmi les perspectives d'amélioration envisagées, nous proposons :

- Intégrer l'API TikTok afin de permettre la publication automatique des offres sur cette plateforme et renforcer la visibilité de l'agence ;
- Ajouter PayPal comme nouveau moyen de paiement, en complément de Chargily ePay, pour offrir plus de flexibilité aux utilisateurs ;
- Mettre en place l'internationalisation du site (multi-langues), afin de rendre l'application accessible à un public étranger ;
- Intégrer la recherche des villes directement dans l'application à l'aide de l'API site-rapid, pour récupérer automatiquement les identifiants des destinations et enrichir la base de données des villes proposées ;
- Gérer les types et les tarifs des visas et des assurances, afin de fournir des informations précises et à jour lors des demandes ;
- Permettre à l'administrateur et à l'agent de créer, modifier ou supprimer les types de visas et d'assurances selon les besoins.
- Le développement d'une application mobile pour permettre aux utilisateurs d'accéder facilement aux services depuis leur smartphone, à tout moment.

# Annexes

## Diagramme de cas d'utilisation global

Ce diagramme illustre les interactions entre les utilisateurs (utilisateur-inscrit, clients, agents, administrateurs) et les différentes fonctionnalités offertes par la plateforme.

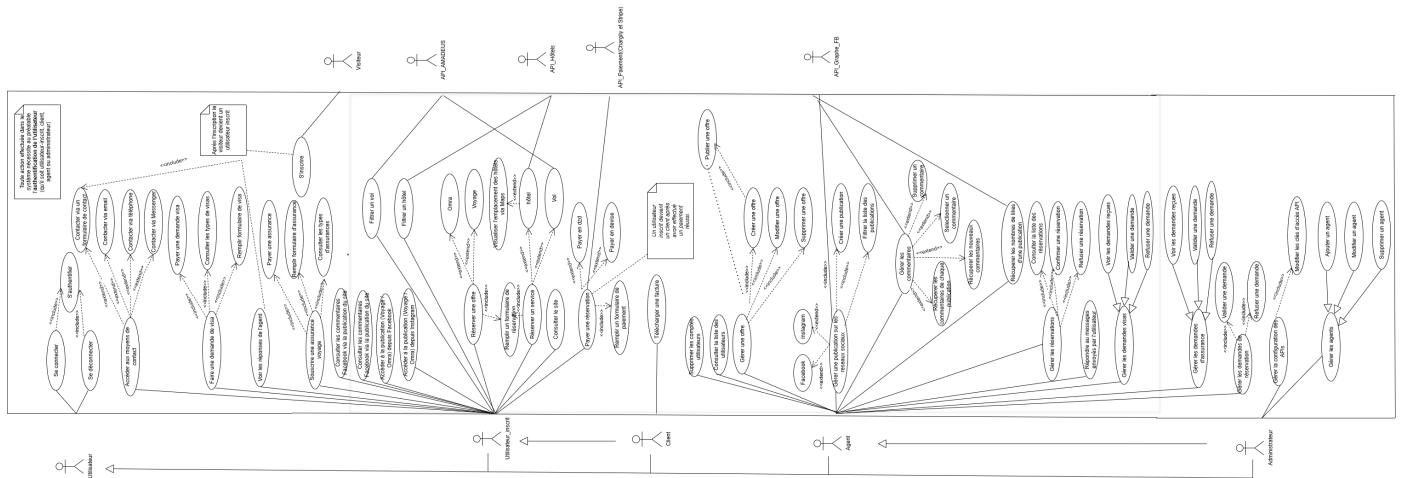


FIGURE 4.1 – Diagramme de cas d'utilisation global de l'application

# Diagramme de classe global

Ce diagramme présente les principales entités métier, leurs attributs et les relations entre elles dans l'application. Il constitue la base du modèle de données.

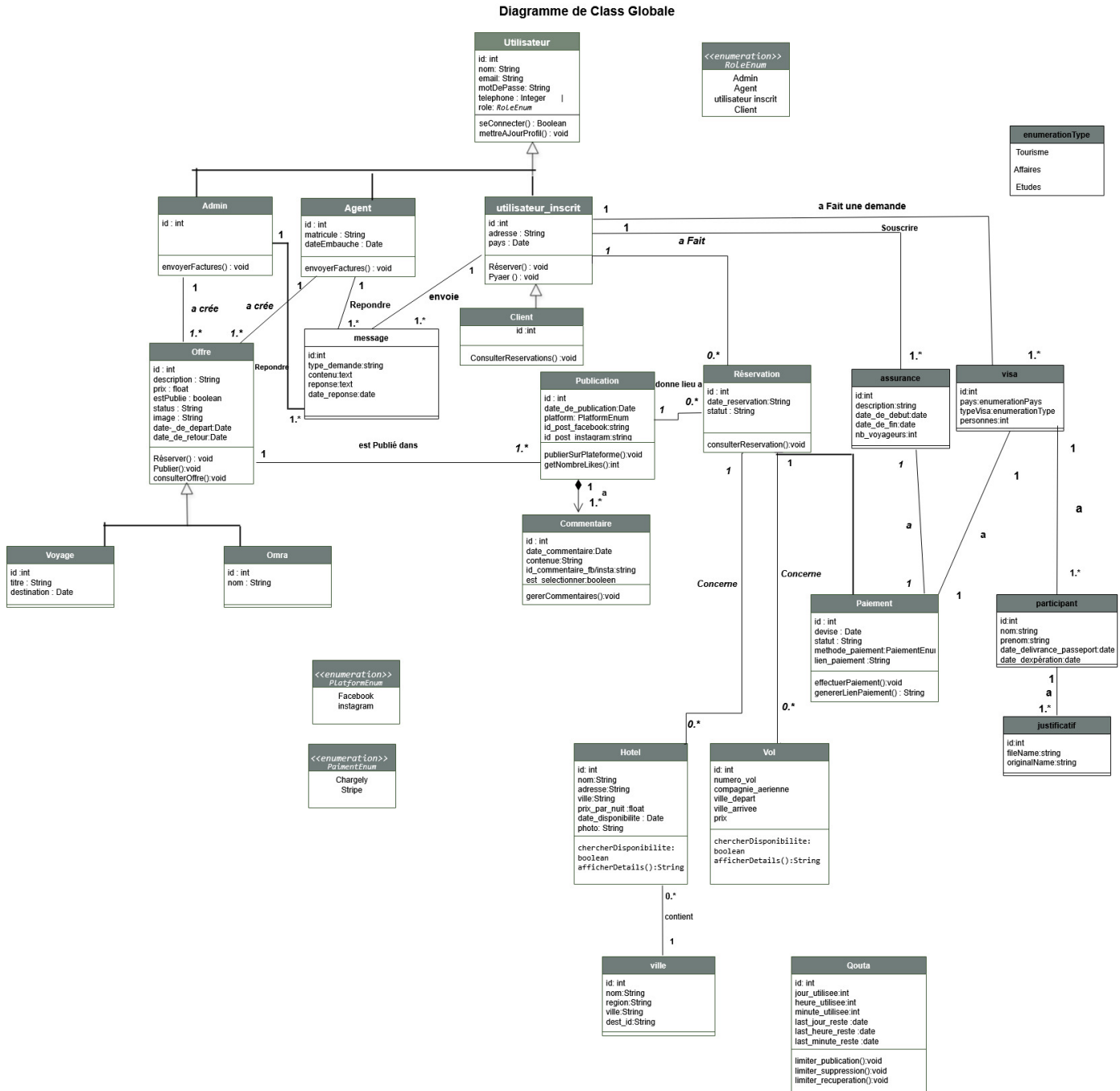


FIGURE 4.2 – Diagramme de classe global de l'application

# Bibliographie

- [1] ORGANISATION MONDIALE DU TOURISME (OMT). *Glossaire des termes du tourisme*. <https://www.unwto.org/fr/glossaire-de-tourisme>. Consulté le 20 février 2025.
- [2] ESLCSA. *Quels sont les types de tourisme ?* <https://www.eslcsa.ma/blog/quels-sont-les-types-de-tourisme>. Consulté le 20 février 2025.
- [3] GOOGLE. *Agences de voyages à Béjaïa*. <https://www.google.com/search?client=firefox-b-d&q=agences++de+voyages+bejaia>. Consulté le 6 mars 2025.
- [4] AFALENCE. *Wappalyzer*. <https://www.afalence.com/articles/wappalyzer>. Consulté le 13 mars 2025.
- [5] SITE-ANALYZER. *Conditions d'utilisation*. <https://www.site-analyzer.com/fr/terms-of-service>. Consulté le 17 mars 2025.
- [6] Véronique Messenger ROTA. *Gestion de projet agile : avec Scrum, Lean, extreme Programming...* Dunod, 2010.
- [7] Nasser KETTANI. *De Merise à UML*. Éditions Eyrolles, 2001.
- [8] POSTMAN. *Your Complete API Platform*. <https://www.postman.com/#\protect\protect\leavevmode@ifvmode\kern+.2222em\relax~\protect\protect\leavevmode@ifvmode\kern+.2222em\relaxtext=Your%20Complete%20API%20Platform%2C%20From,Design%20to%20Delivery>. Consulté le 6 mai 2025.
- [9] GITHUB. *À propos de GitHub et Git*. <https://docs.github.com/fr/get-started/start-your-journey/about-github-and-git/#\protect\protect\leavevmode@ifvmode\kern+.2222em\relax~\protect\protect\leavevmode@ifvmode\kern+.2222em\relaxtext=GitHub%20est%20une%20plateforme%20bas%C3%A9e,d%27autres%20pour%20%C3%A9crire%20du%20code>. Consulté le 6 mai 2025.
- [10] Meta for DEVELOPERS. *Connexion et intégration des services Meta*. <https://business.facebook.com/business/loginpage>. Consulté le 6 mai 2025.
- [11] PHPMYADMIN. *phpMyAdmin – outil de gestion de base de données*. <https://www.phpmyadmin.net/#\protect\protect\leavevmode@ifvmode\kern+.2222em\relax~\protect\protect\leavevmode@ifvmode\kern+.2222em\relaxtext=phpMyAdmin%20is%20a%20free%20software,directly%20execute%20any%20SQL%20statement>. Consulté le 6 mai 2025.
- [12] Visual Studio CODE. *Éditeur de code open source*. <https://framalibre.org/content/visual-studio-code>. Consulté le 6 mai 2025.
- [13] REACT. *Getting Started – React Documentation*. <https://legacy.reactjs.org/docs/getting-started.html>. Consulté le 6 mai 2025.

- [14] BOOTSTRAP. *Everything You Need to Know About the Bootstrap Framework*. <https://www.quickstart.com/blog/creative-and-design/everything-you-need-to-know-about-the-bootstrap-framework/#\protect\protect\leavevmode@ifvmode\kern+.2222em\relax~\protect\protect\leavevmode@ifvmode\kern+.2222em\relaxtext=Bootstrap%20is%20a%20popular%20open,first%20websites>. Consulté le 6 mai 2025.
- [15] NODE.JS. *À propos de Node.js*. <https://nodejs.org/en/about>. Consulté le 6 mai 2025.
- [16] ÉRIC SARRION. *Programmation avec Node.js, Express.js et MongoDB*. Eyrolles, 2014. ISBN : 978-2-212-13994-5.
- [17] ORACLE. *Qu'est-ce que MySQL ?* <https://www.oracle.com/africa-fr/mysql/what-is-mysql/#\protect\protect\leavevmode@ifvmode\kern+.2222em\relax~\protect\protect\leavevmode@ifvmode\kern+.2222em\relaxtext=MySQL%20est%20un%20syst%C3%A8me%20de,le%20trouverez%20au%20c%C5%93ur%20des>. Consulté le 6 mai 2025.
- [18] SEQUELIZE. *Documentation officielle de Sequelize ORM*. <https://sequelize.org/#\protect\protect\leavevmode@ifvmode\kern+.2222em\relax~\protect\protect\leavevmode@ifvmode\kern+.2222em\relaxtext=Sequelize>. Consulté le 6 mai 2025.
- [19] XAMPP. *Apache Friends – XAMPP*. <https://www.apachefriends.org/fr/index.html/#\protect\protect\leavevmode@ifvmode\kern+.2222em\relax~\protect\protect\leavevmode@ifvmode\kern+.2222em\relaxtext=XAMPP%20est%20une%20distribution%20Apache,%C3%A0%20installer%20et%20%C3%A0%20utiliser>. Consulté le 6 mai 2025.
- [20] Mozilla Developer NETWORK. *AJAX – Glossaire*. <https://developer.mozilla.org/en-US/docs/Glossary/AJAX>. Consulté le 6 mai 2025.
- [21] Meta DEVELOPERS. *Graph API Overview*. <https://developers.facebook.com/docs/graph-api/overview>. Consulté le 21 juin 2025.
- [22] Inc. META PLATFORMS. *Instagram Platform - Meta for Developers*. <https://developers.facebook.com/docs/instagram-platform>. Consulté en juin 2025. 2024.
- [23] Inc. META PLATFORMS. *Graph API Reference – Page*. <https://developers.facebook.com/docs/graph-api/reference/page>. Consulté en juin 2025. 2025.
- [24] Inc. META PLATFORMS. *Access Tokens - Facebook Login*. <https://developers.facebook.com/docs/facebook-login/guides/access-tokens>. Consulté en juin 2025. 2025.
- [25] Inc. META PLATFORMS. *Graph API Reference - Page Photos*. <https://developers.facebook.com/docs/graph-api/reference/page/photos/>. Consulté en juin 2025. 2025.
- [26] Inc. META PLATFORMS. *Graph API Reference - Comment*. <https://developers.facebook.com/docs/graph-api/reference/comment/>. Consulté en juin 2025. 2025.

- [27] Inc. META PLATFORMS. *Instagram Graph API Reference – Instagram Media*. <https://developers.facebook.com/docs/instagram-platform/reference/instagram-media/>. Consulté en juin 2025. 2025.
- [28] Inc. META PLATFORMS. *Comment Moderation – Instagram Graph API*. <https://developers.facebook.com/docs/instagram-platform/instagram-api-with-facebook-login/comment-moderation/>. Consulté en juin 2025. 2024.
- [29] Inc. META PLATFORMS. *Rate Limiting - Facebook Graph API*. <https://developers.facebook.com/docs/graph-api/overview/rate-limiting/>. Consulté en juin 2025. 2025.
- [30] Inc. META PLATFORMS. *Content Publishing - Instagram Platform*. <https://developers.facebook.com/docs/instagram-platform/content-publishing/>. Consulté en juin 2025. 2025.
- [31] Inc. META PLATFORMS. *Handle Errors – Graph API*. <https://developers.facebook.com/docs/graph-api/guides/error-handling/>. Consulté en juin 2025. 2025.
- [32] NGROK. *Ngrok Documentation*. <https://ngrok.com/docs>. Consulté en juin 2025. 2025.
- [33] STRIPE. *Stripe API Documentation*. <https://docs.stripe.com/api>. Consulté à plusieurs reprises entre mars et juin 2025.
- [34] CHARGILY. *Chargily Pay API – Documentation développeur*. <https://dev.chargily.com/pay-v2/introduction>. Consulté à plusieurs reprises entre mars et juin 2025.

# Résumé

## Résumé

Dans le cadre de notre projet de fin d'études, nous avons conçu et développé ZIGUADE TOUR, une application web moderne et intelligente dédiée aux services touristiques et de voyage. Elle s'adresse à un large public, aussi bien national qu'international, En proposant une plateforme numérique centralisée pour la réservation de voyages organisés, d'hôtels, de vols, de Omras, ainsi que la gestion des assurances et des demandes de visa. Le projet repose sur une architecture bien définie, avec une séparation claire entre le front-office, destiné aux utilisateurs finaux, et le back-office, réservé aux administrateurs et agents. L'application permet une gestion sécurisée des comptes, des offres et des réservations, tout en assurant une expérience utilisateur fluide et intuitive. Nous avons intégré plusieurs services externes, notamment l'API Amadeus pour les vols, Chargily ePay pour les paiements locaux (cartes CIB et Dahabia), et Stripe pour les paiements internationaux. Les API de Facebook et d'Instagram permettent la publication d'offres par les administrateurs, que l'application récupère et affiche automatiquement aux utilisateurs. Le développement a été mené en suivant la méthode Scrum, garantissant une organisation agile et efficace. ZIGUADE TOUR se positionne comme une solution digitale complète, innovante et évolutive pour moderniser et simplifier l'accès aux services touristiques.

## Abstract

As part of our final year project, we designed and developed ZIGUADE TOUR, a modern and intelligent web application dedicated to tourism and travel services. It targets a broad audience, both nationally and internationally, by offering a centralized digital platform for booking organized trips, hotels, flights, Omras, as well as managing insurance and visa applications. The project is based on a well-defined architecture with a clear separation between the front office, intended for end users, and the back office, reserved for administrators and agents. The application enables secure management of accounts, offers, and bookings while ensuring a smooth and intuitive user experience. We integrated several external services, including the Amadeus API for flights, Chargily ePay for local payments (CIB and Dahabia cards), and Stripe for international payments. The Facebook and Instagram APIs allow administrators to publish offers, which the application then retrieves and displays automatically to users. The development followed the Scrum methodology, ensuring an agile and efficient organization.

ZIGUADE TOUR positions itself as a complete, innovative, and scalable digital solution for modernizing and simplifying access to tourism services.