

## **Remerciements**

Je remercie DIEU qui m'a ouvert toutes les portes et facilité toute chose, sans lui rien de bon n'aurait été possible.

J'exprime ma gratitude à mon encadrante, Mme NOUICER Amina, pour sa disponibilité et son implication tout au long de ce travail.

Mes remerciements vont également à Mr BEKKA Reda pour m'avoir proposé le présent thème et pour m'avoir aidé tout au long de mon stage.

Je tiens à remercier également les membres du jury pour avoir consacré leur temps à examiner ce mémoire, pour l'intérêt qu'ils ont porté à mon travail et pour leurs contributions à l'enrichir.

Enfin, je tiens à remercier toutes les personnes qui, de près ou de loin, ont contribué à la réalisation de ce travail.

## *Dédicaces*

*A la mémoire de mon père Kamel, Allah irahmou.*

*Je dédie ce modeste travail à :*

❖ *Ma mère qui m'a soutenu et accompagné tout au long de ma vie et m'a offert inconditionnellement les conditions et encouragements pour réussir.*

❖ *Tous mes oncles et tantes maternels et paternels.*

❖ *Mes cousins et cousines maternels et paternels.*

❖ *Mes enseignants.*

❖ *Mes amis.*

## Table des matières

<b>Introduction générale .....</b>	<b>1</b>
<b>Chapitre 1 Étude de l'existant .....</b>	<b>3</b>
<b>1.1 Introduction .....</b>	<b>3</b>
<b>1.2 Présentation de l'organisme d'accueil BigNova .....</b>	<b>3</b>
<b>1.2.1 Les services de BigNova.....</b>	<b>3</b>
<b>1.2.2 Organigramme de la boîte de développement BigNova .....</b>	<b>4</b>
<b>1.3 Motivations .....</b>	<b>5</b>
<b>1.3.1 Applications existantes .....</b>	<b>6</b>
<b>1.4 Présentation de l'application.....</b>	<b>7</b>
<b>1.4.1 Objectifs.....</b>	<b>7</b>
<b>1.4.2 Avantages.....</b>	<b>7</b>
<b>1.5 Conclusion.....</b>	<b>8</b>
<b>Chapitre 2 Démarche de développement, langage de modélisation et technologies utilisées</b>	<b>10</b>
<b>2.1 Introduction .....</b>	<b>10</b>
<b>2.2 La méthode Scrum .....</b>	<b>10</b>
<b>2.2.1 Rôles de la méthode Scrum .....</b>	<b>10</b>
<b>2.2.2 Artefacts de Scrum .....</b>	<b>11</b>
<b>2.2.3 Les événements de la méthode Scrum.....</b>	<b>12</b>
<b>2.3 Langage de modélisation .....</b>	<b>15</b>
<b>2.4 Technologies utilisées .....</b>	<b>16</b>
<b>2.4.1 Front-end .....</b>	<b>16</b>
<b>2.4.2 Back-end .....</b>	<b>17</b>
<b>2.5 Conclusion.....</b>	<b>18</b>
<b>Chapitre 3 Réalisation du projet selon la démarche SCRUM.....</b>	<b>19</b>
<b>3.1 Identification des acteurs.....</b>	<b>19</b>
<b>3.1.1 Administrateur .....</b>	<b>19</b>
<b>3.1.2 Partenaire .....</b>	<b>20</b>
<b>3.1.3 Visiteur anonyme .....</b>	<b>20</b>
<b>3.1.4 Client .....</b>	<b>20</b>
<b>3.2 Répartition des rôles .....</b>	<b>21</b>
<b>3.3 User stories.....</b>	<b>21</b>
<b>3.4 Diagramme de contexte .....</b>	<b>23</b>
<b>3.5 Diagramme de cas d'utilisation global .....</b>	<b>23</b>

<b>3.6</b>	<b>Product Backlog</b> .....	<b>24</b>
<b>3.7</b>	<b>Planification des sprints</b> .....	<b>25</b>
<b>3.8</b>	<b>Sprint 1</b> .....	<b>26</b>
<b>3.8.1</b>	<b>Descriptions textuelles</b> .....	<b>26</b>
<b>3.8.2</b>	<b>Diagramme de cas d'utilisation du sprint 1</b> .....	<b>28</b>
<b>3.8.3</b>	<b>Diagrammes de séquence</b> .....	<b>28</b>
<b>3.8.4</b>	<b>Diagramme de classes participantes</b> .....	<b>31</b>
<b>3.8.5</b>	<b>Interfaces utilisateur</b> .....	<b>31</b>
<b>3.9</b>	<b>Sprint 2</b> .....	<b>37</b>
<b>3.9.1</b>	<b>Descriptions textuelles</b> .....	<b>38</b>
<b>3.9.2</b>	<b>Diagramme de cas d'utilisation du sprint 2</b> .....	<b>44</b>
<b>3.9.3</b>	<b>Diagrammes de séquence</b> .....	<b>45</b>
<b>3.9.4</b>	<b>Diagramme de classes participantes</b> .....	<b>50</b>
<b>3.9.5</b>	<b>Interfaces utilisateur</b> .....	<b>50</b>
<b>3.10</b>	<b>Sprint 3</b> .....	<b>54</b>
<b>3.10.1</b>	<b>Descriptions textuelles</b> .....	<b>54</b>
<b>3.10.2</b>	<b>Diagramme de cas d'utilisation du sprint 3</b> .....	<b>57</b>
<b>3.10.3</b>	<b>Diagrammes de séquence</b> .....	<b>58</b>
<b>3.10.4</b>	<b>Diagramme de classes participantes</b> .....	<b>59</b>
<b>3.10.5</b>	<b>Interfaces utilisateur</b> .....	<b>60</b>
<b>3.11</b>	<b>Sprint 4</b> .....	<b>61</b>
<b>3.11.1</b>	<b>Descriptions textuelles</b> .....	<b>61</b>
<b>3.11.2</b>	<b>Diagramme de cas d'utilisation du sprint 4</b> .....	<b>64</b>
<b>3.11.3</b>	<b>Diagrammes de séquence</b> .....	<b>64</b>
<b>3.11.4</b>	<b>Diagrammes de classes participantes</b> .....	<b>66</b>
<b>3.11.5</b>	<b>Interfaces utilisateur</b> .....	<b>67</b>
<b>3.12</b>	<b>Dictionnaire de données</b> .....	<b>68</b>
<b>3.13</b>	<b>Diagramme de classes global :</b> .....	<b>70</b>
<b>3.14</b>	<b>Conclusion</b> .....	<b>71</b>
<b>Conclusion Générale</b> .....		<b>72</b>
<b>Bibliographie</b> .....		<b>73</b>

## Liste des acronymes

---

<b>IA</b>	→	Intelligence Artificielle.
<b>SaaS</b>	→	Software As A Service
<b>IT</b>	→	Information Technology
<b>CTO</b>	→	Chief Technology Officer
<b>B2B</b>	→	Business To Business
<b>DJ</b>	→	Disk Jockey
<b>CIB</b>	→	Carte InterBancaire
<b>UML</b>	→	Unified Modeling Language
<b>MVC</b>	→	Model View Controller
<b>PO</b>	→	Product Owner
<b>HTML</b>	→	Hyper Text Markup Language
<b>CSS</b>	→	Cascading Style Sheets
<b>JS</b>	→	JavaScript
<b>SEO</b>	→	Search Engine Optimization
<b>SSG</b>	→	Static Site Generation
<b>SSR</b>	→	Server Side Rendering
<b>MongoDB</b>	→	Humongous DataBase
<b>DOM</b>	→	Document Object Model
<b>NPM</b>	→	Node Package Manager
<b>API</b>	→	Application Programming Interface

**BDD** —————> Base De Données

**NoSQL** —————> Not Only Structured Query Language

**JSON** —————> JavaScript Object Notation

**BSON** —————> Binary JavaScript Object Notation

**IOT** —————> Internet Of Things

**MEAN** —————> MongoDB ExpressJS Angular NodeJS

**MERN** —————> MongoDB ExpressJS React NodeJS

## Table des figures

---

<b>Figure 1: Organigramme BigNova</b> .....	5
<b>Figure 2: Cycle de vie de la méthode Scrum</b> .....	12
<b>Figure 3: Diagramme de contexte</b> .....	23
<b>Figure 4: Diagramme de cas d'utilisation global</b> .....	24
<b>Figure 5 : Diagramme de Gantt</b> .....	26
<b>Figure 6 : Diagramme de cas d'utilisation du sprint 1</b> .....	28
<b>Figure 7: Diagramme de séquence du cas d'utilisation S'inscrire</b> .....	29
<b>Figure 8: Diagramme de séquence du cas d'utilisation S'authentifier</b> .....	30
<b>Figure 9 : Diagramme de classes participantes du sprint 1</b> .....	31
<b>Figure 10: Interface d'accueil Client</b> .....	32
<b>Figure 11: Interface d'accueil Administrateur</b> .....	32
<b>Figure 12: Interface d'accueil Partenaire</b> .....	33
<b>Figure 13: Interface S'inscrire pour le client</b> .....	34
<b>Figure 14: Interface S'inscrire pour le partenaire et l'administrateur</b> .....	35
<b>Figure 15: Interface S'authentifier pour le client</b> .....	36
<b>Figure 16: Interface S'authentifier pour le partenaire et l'administrateur</b> .....	37
<b>Figure 17: Diagramme de cas d'utilisation du sprint 2</b> .....	45
<b>Figure 18: Diagramme de séquence du cas d'utilisation Ajouter un client</b> .....	46
<b>Figure 19 Diagramme de séquence du cas d'utilisation Modifier un client</b> .....	46
<b>Figure 20 Diagramme de séquence du cas d'utilisation Supprimer un client</b> .....	47
<b>Figure 21: Diagramme de séquence du cas d'utilisation Ajouter un partenaire</b> .....	47
<b>Figure 22: Diagramme de séquence du cas d'utilisation Modifier un partenaire</b> .....	48
<b>Figure 23: Diagramme de séquence du cas d'utilisation Supprimer un partenaire</b> .....	48
<b>Figure 24: Diagramme de séquence du cas d'utilisation Ajouter un service</b> .....	49
<b>Figure 25: Diagramme de séquence du cas d'utilisation Modifier un service</b> .....	49
<b>Figure 26: Diagramme de séquence du cas d'utilisation Supprimer un service</b> .....	50
<b>Figure 27: Diagramme de classes participantes du sprint 2</b> .....	50
<b>Figure 28: Interface Ajouter un partenaire</b> .....	51
<b>Figure 29: Interface Modifier un partenaire</b> .....	51
<b>Figure 30: Interface Supprimer un partenaire</b> .....	52
<b>Figure 31: Interface Ajouter un client</b> .....	52
<b>Figure 32: Interface Modifier un client</b> .....	52
<b>Figure 33: Interface Supprimer un client</b> .....	53
<b>Figure 34: Interface Ajouter un service</b> .....	53
<b>Figure 35: Interface Modifier un service</b> .....	53
<b>Figure 36: Interface Supprimer un service</b> .....	54
<b>Figure 37 : Diagramme de cas d'utilisation du sprint 3</b> .....	57
<b>Figure 38: Diagramme de séquence du cas d'utilisation Ajouter une réservation</b> .....	58
<b>Figure 39: Diagramme de séquence du cas d'utilisation Modifier une réservation</b> .....	58
<b>Figure 40: Diagramme de séquence du cas d'utilisation Supprimer une réservation</b> .....	59

<b>Figure 41: Diagramme de classes participantes du sprint 3</b> .....	59
<b>Figure 42: Interface Ajouter une réservation</b> .....	60
<b>Figure 43: Interface Modifier une réservation</b> .....	60
<b>Figure 44: Interface Supprimer une réservation</b> .....	60
<b>Figure 45: Diagramme de cas d'utilisation du sprint 4</b> .....	64
<b>Figure 46: Diagramme de séquence du cas d'utilisation Ajouter un événement</b> .....	65
<b>Figure 47: Diagramme de séquence du cas d'utilisation Modifier un événement</b> .....	65
<b>Figure 48: Diagramme de séquence du cas d'utilisation Supprimer un événement</b> .....	66
<b>Figure 49: Diagramme de séquence du cas d'utilisation Annuler un événement</b> .....	66
<b>Figure 50: Diagramme de classes participantes du sprint 4</b> .....	67
<b>Figure 51: Interface Ajouter un événement</b> .....	67
<b>Figure 52: Interface Modifier un événement</b> .....	68
<b>Figure 53: Interface Supprimer/Annuler un événement</b> .....	68
<b>Figure 54: Diagramme de classes global</b> .....	70

## Liste des Tableaux

---

<b>Tableau 1 : Répartition des rôles.....</b>	<b>21</b>
<b>Tableau 2 : User stories .....</b>	<b>21</b>
<b>Tableau 3 : Product Backlog.....</b>	<b>25</b>
<b>Tableau 4 : Description textuelle du cas d'utilisation S'inscrire.....</b>	<b>27</b>
<b>Tableau 5 : Description textuelle du cas d'utilisation S'authentifier.....</b>	<b>27</b>
<b>Tableau 6 : Description textuelle du cas d'utilisation Ajouter un client .....</b>	<b>38</b>
<b>Tableau 7 : Description textuelle du cas d'utilisation Modifier un client.....</b>	<b>39</b>
<b>Tableau 8 : Description textuelle du cas d'utilisation Supprimer un client.....</b>	<b>40</b>
<b>Tableau 9 : Description textuelle du cas d'utilisation Ajouter un partenaire.....</b>	<b>40</b>
<b>Tableau 10 : Description textuelle du cas d'utilisation Modifier un partenaire .....</b>	<b>41</b>
<b>Tableau 11 : Description textuelle du cas d'utilisation Supprimer un partenaire .....</b>	<b>42</b>
<b>Tableau 12 : Description textuelle du cas d'utilisation Ajouter un service .....</b>	<b>42</b>
<b>Tableau 13 : Description textuelle du cas d'utilisation Modifier un service .....</b>	<b>43</b>
<b>Tableau 14 : Description textuelle du cas d'utilisation Supprimer un service.....</b>	<b>44</b>
<b>Tableau 15 : Description textuelle du cas d'utilisation Ajouter une réservation.....</b>	<b>55</b>
<b>Tableau 16 : Description textuelle du cas d'utilisation Modifier une réservation.....</b>	<b>56</b>
<b>Tableau 17 : Description textuelle du cas d'utilisation Supprimer une réservation .....</b>	<b>56</b>
<b>Tableau 18 : Description textuelle du cas d'utilisation Ajouter un événement.....</b>	<b>61</b>
<b>Tableau 19 Description textuelle du cas d'utilisation Modifier un événement .....</b>	<b>62</b>
<b>Tableau 20 : Description textuelle du cas d'utilisation Supprimer un événement .....</b>	<b>62</b>
<b>Tableau 21 : Description textuelle du cas d'utilisation Annuler un événement.....</b>	<b>63</b>



## Introduction générale

---

Dans le cadre de notre formation, nous avons eu l'opportunité d'effectuer un stage au sein de **BigNova**, une entreprise dynamique spécialisée dans le développement de solutions numériques sur mesure. Cette immersion en entreprise m'a permis de mettre en pratique mes compétences techniques tout en découvrant les méthodes agiles utilisées dans des projets concrets.

Durant ce stage, j'ai été chargé de concevoir et de développer une **plateforme web dédiée à l'organisation de services événementiels**. Cette application a pour ambition de **faciliter l'organisation d'événements privés** (comme les mariages, anniversaires, fêtes familiales) et **d'accompagner les entreprises privées** dans la **gestion de leurs événements professionnels** (séminaires, conférences, lancements de produits, etc.).

Parmi les **fonctionnalités clés** de la plateforme, on retrouve :

- Un **catalogue interactif de prestataires de services** (traiteurs, photographes, animateurs, etc.),
- Une **recherche personnalisée** selon les besoins de l'utilisateur,
- Un système complet de **gestion des réservations**,
- La **création et gestion d'événements** à travers une interface intuitive,
- Une **marketplace** permettant d'acheter ou de louer des produits liés aux événements.

Le présent rapport se structure autour de **trois chapitres complémentaires**.

Le **premier chapitre** est consacré à la **présentation de l'entreprise BigNova** ainsi qu'à une description approfondie de l'application web que j'ai développée, incluant ses objectifs fonctionnels, son public cible et ses principales spécificités.

Dans le **deuxième chapitre**, j'aborde la **méthode agile Scrum**, qui a été adoptée pour encadrer le déroulement du projet. Cette approche itérative et collaborative m'a permis de structurer efficacement les phases du développement, tout en assurant une réactivité face aux ajustements fonctionnels.

Le **troisième chapitre** présente en détail le **processus de conception et de réalisation de la plateforme**, en s'appuyant sur les sprints Scrum définis précédemment. Il retrace l'évolution du projet, les choix techniques effectués ainsi que les différentes étapes du développement, de l'analyse des besoins jusqu'à la mise en œuvre des fonctionnalités principales.

Sur le plan technologique, j'ai opté pour une architecture **full JavaScript** moderne. Le **frontend** a été développé en **HTML5**, **CSS3**, **React** et **Next.js**, pour offrir une interface fluide et réactive. Le **backend** repose sur **Node.js** avec le framework **Express.js**, tandis que les données sont stockées dans une base **MongoDB**, choisie pour sa souplesse dans la gestion de documents structurés.

Ce projet m'a offert une expérience complète, à la fois sur les aspects techniques, organisationnels et méthodologiques, me préparant ainsi aux exigences du monde professionnel et aux pratiques actuelles du développement web.

# Chapitre 1 Étude de l'existant

## 1.1 Introduction

Ce chapitre s'inscrit dans **la phase d'étude de l'existant**, c'est-à-dire l'analyse des **besoins réels du terrain** et des **dysfonctionnements observés** dans le domaine de l'organisation d'événements. L'existant, dans ce contexte, correspond aux pratiques actuelles, aux outils limités utilisés par les particuliers et les professionnels, ainsi qu'aux difficultés récurrentes rencontrées : manque de visibilité des prestataires, dispersion des services, complexité de la gestion logistique, absence de planification centralisée, et paiements peu adaptés. Ces constats révèlent un **besoin concret de centraliser, simplifier et digitaliser** l'organisation d'événements à travers une solution complète, intuitive et adaptée.

Ainsi, le chapitre débute par la présentation de l'entreprise BigNova, organisme d'accueil du stage (section 1.2), puis décrit les différents services qu'elle propose (section 1.3). Il présente ensuite l'organisation interne de l'entreprise via son organigramme (section 1.4). La section suivante est consacrée à l'analyse des problèmes identifiés sur le terrain, à l'origine du projet d'application (section 1.5). Enfin, l'application envisagée est détaillée à travers ses objectifs, ses fonctionnalités clés et les réponses concrètes qu'elle apporte aux besoins relevés (section 1.6), avant de conclure sur les enseignements tirés de cette étude de l'existant (section 1.7).

## 1.2 Présentation de l'organisme d'accueil BigNova

BigNova (BigNova, s.d.), est une boîte de développement créative et réactive basée à Béjaïa et à Paris, France. Expert en création de sites Internet, applications web et mobiles sur mesure, web design, graphic design, conseils et optimisation digitale, BigNova est une boîte fondée et dirigée sur des technologies du web, mobile et de design. BigNova est notamment compétente en :

- Création d'outils d'IA.
- Développement d'applications mobiles.
- Devops et cloud computing.
- Développement de solutions SaaS.
- Consulting et formations IT.

### 1.2.1 Les services de BigNova

#### 1.2.1.1 Développement web et mobile

BigNova offre des solutions sur mesure pour sites web et applications web adaptés aux besoins spécifiques de chaque entreprise.

### **1.2.1.2 Développement en régie**

BigNova propose des développeurs répondant aux besoins variés des entreprises en termes de compétences techniques.

### **1.2.1.3 DevOps et cloud :**

BigNova optimise la synergie entre le cloud et DevOps pour améliorer les performances d'hébergement, renforçant les pratiques DevOps dans un environnement multi-cloud. BigNova est une entreprise de développement informatique qui se consacre à fournir des solutions technologiques innovantes pour les entreprises de toutes tailles. Elle est rapidement devenue un leader dans l'industrie grâce à son expertise technique, sa passion pour l'innovation et son engagement envers la satisfaction du client.

L'objectif principal de BigNova est d'aider les entreprises à relever les défis technologiques auxquels elles sont confrontées en proposant des solutions personnalisées qui répondent à leurs besoins spécifiques. Que ce soit pour la création de sites web, design web / mobile, la création d'applications mobiles, devOps, la conception, cloud computing.

BigNova dispose d'une équipe de développeurs hautement qualifiés et expérimentés qui possèdent une expertise dans une variété de langages de programmation, de frameworks et de technologies.

## **1.2.2 Organigramme de la boîte de développement BigNova**

### **1.2.2.1 Gérant et Co-gérante**

Ils assurent la direction générale de l'entreprise.

### **1.2.2.2 CTO (Chief Technologies Officer)**

Il est responsable de la stratégie technique, il supervise l'ensemble des projets de développement et les équipes techniques.

### **1.2.2.3 Community manager**

Il chargé de la communication, de la gestion de la communauté et de la promotion des activités (Formation et développement).

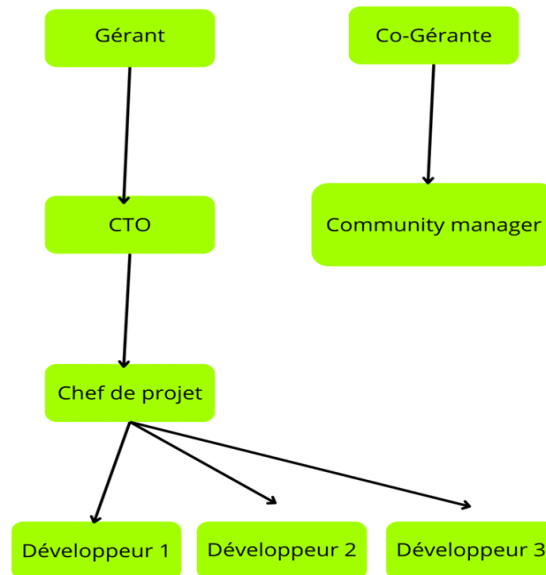
### **1.2.2.4 Chef de projet**

Il organise et coordonne les projets techniques, sous la direction du CTO.

### **1.2.2.5 Développeurs :**

Ils réalisent les développements techniques nécessaires aux projets.

La figure 1 représente la hiérarchie de la boîte BigNova.



**Figure 1: Organigramme BigNova**

### 1.3 Motivations

Développer une application dédiée aux services pour événements et fêtes répond à un besoin croissant d’optimisation et de simplification dans l’organisation d’événements privés et professionnels. Aujourd’hui, de nombreuses personnes rencontrent des difficultés pour trouver les bons prestataires, comparer les offres et gérer efficacement les différentes étapes de leur événement. Une application centralisant ces services offre une solution clé en main, permettant aux utilisateurs de gagner du temps et de réduire le stress lié à l’organisation.

En outre, le secteur de l’événementiel est en pleine expansion, notamment avec la digitalisation des services. Une application bien conçue permettrait de mettre en relation facilement les clients et les prestataires, assurant une meilleure visibilité aux professionnels du domaine et une expérience utilisateur améliorée. Grâce à des fonctionnalités telles que la recherche avancée, la gestion des réservations et paiements sécurisés, ainsi que la planification d’événements, cette solution garantirait une organisation fluide et efficace.

De plus, en intégrant un module de marketplace, l’application pourrait répondre à des besoins annexes comme l’achat de décorations, de cadeaux ou d’accessoires spécifiques, créant ainsi un écosystème complet dédié aux événements. En élargissant l’offre à des services professionnels (Séminaires, conférences, etc...), l’application pourrait également capter un marché Business To Business (B2B), renforçant ainsi son potentiel de croissance et de rentabilité.

Enfin, une telle plateforme permettrait de moderniser le secteur en apportant plus de transparence, de flexibilité et de personnalisation, tout en soutenant les acteurs locaux en leur offrant un canal digital pour promouvoir leurs services.

### 1.3.1 Applications existantes

Les applications web sur le thème de l'événementiel en Algérie ne sont pas généralistes, voici quelques applications

#### 1.3.1.1 Allo Mariage DZ

**Type d'événement :** Mariage

**Application web :** Disponible sur navigateur (site responsive).

**Fonctionnalités :** Répertoire de prestataires pour mariages (salles, robes, traiteurs, etc.), prise de contact.

**Points faibles :**

- Interface peu ergonomique sur certains navigateurs.
- Pas de gestion complète du planning ni de tableau de bord personnalisé.
- Aucune gestion de budget ni de contrats intégrés.

#### 1.3.1.2 Happy Anniv DZ

**Type d'événement :** Anniversaire (enfants et adultes)

**Application web :** Présente sous forme de site vitrine avec formulaire de réservation

**Fonctionnalités :** Propose des packs d'anniversaire (animation, décoration, gâteaux).

**Points faibles :**

- Pas de compte utilisateur ni de tableau de bord pour suivre l'organisation.
- Réservation peu interactive : formulaire basique, sans disponibilité en temps réel.
- Aucun suivi automatisé (emails, rappels, confirmations).

#### 1.3.1.3 Eventia Pro DZ

**Type d'événement :** Conférences & séminaires professionnels

**Application web :** Application en ligne avec inscription des participants

**Fonctionnalités :** Gestion des intervenants, participants, envoi d'invitations, génération de badges.

**Points faibles :**

- Complexité d'usage pour les non-initiés (trop orientée B2B).
- Pas d'intégration avec des outils de visioconférence.
- Design peu moderne, difficilement utilisable sur mobile.

## 1.4 Présentation de l'application

### 1.4.1 Objectifs

#### 1. Faciliter l'organisation d'événements privés

L'organisation d'un événement privé peut être un véritable défi, que ce soit pour un mariage, des fiançailles, un anniversaire ou une réception traditionnelle. Notre plateforme simplifie ce processus en mettant en relation les utilisateurs avec des prestataires qualifiés pour répondre à leurs besoins spécifiques. Grâce à une interface intuitive, chacun peut planifier son événement en toute sérénité, en sélectionnant les meilleurs services disponibles, du choix de la salle à la décoration, en passant par le traiteur et l'animation.

#### 2. Accompagner les entreprises dans la gestion de leurs événements professionnels (Option future)

Les entreprises ont des exigences particulières lorsqu'il s'agit d'organiser des séminaires, des conférences ou des événements de teambuilding. À terme, notre plateforme proposera un module dédié aux événements professionnels, offrant des solutions adaptées aux besoins des entreprises. Cela inclura la gestion des réservations de salles de conférence, la sélection de traiteurs spécialisés, ainsi que des services audiovisuels et logistiques pour garantir le bon déroulement des événements d'affaires.

### 1.4.2 Avantages

#### 1. Catalogue de prestataires

Pour garantir un large choix aux utilisateurs, notre plateforme propose un catalogue complet de prestataires spécialisés dans l'événementiel. Il inclut des organisateurs d'événements, des loueurs de salles, des traiteurs, des pâtisseries, des décorateurs, des photographes, des vidéastes, des animateurs (DJ, musiciens, etc...), ainsi que des fournisseurs de matériel (Mobilier, éclairage, etc...). Cette diversité permet de trouver facilement les professionnels adaptés à chaque type d'événement.

#### 2. Recherche personnalisée

Trouver le bon prestataire peut être complexe. C'est pourquoi notre moteur de recherche avancée permet aux utilisateurs d'affiner leur sélection selon plusieurs critères : budget, localisation, services souhaités et préférences spécifiques. De plus notre système intelligent propose des suggestions basées sur les tendances du marché et les préférences des utilisateurs, facilitant ainsi la prise de décision.

#### 3. Gestion des réservations et paiements

Notre plateforme intègre un système de réservation en ligne permettant aux utilisateurs de réserver directement les services souhaités. Pour une expérience fluide et sécurisée, plusieurs options de paiement sont disponibles, incluant des solutions adaptées au marché

Algérien comme CIB et Edahabia, ainsi que des méthodes internationales telles que PayPal et les cartes bancaires.

#### 4. **Création et gestion d'événements**

Afin d'aider les utilisateurs à organiser leur événement de manière efficace, notre plateforme propose un planificateur intégré. Celui-ci permet de créer un planning détaillé, de suivre l'état des réservations et d'assurer la coordination avec les prestataires engagés. Les organisateurs peuvent ainsi mieux gérer leurs tâches et éviter les imprévus.

#### 5. **Marketplace pour produits associés**

En plus des services, notre plateforme mettra à disposition une marketplace permettant aux utilisateurs d'acheter des articles liés aux événements. Cela inclut des cadeaux personnalisés, des tenues traditionnelles, des éléments de décoration et bien d'autres produits essentiels pour réussir une célébration.

#### 6. **Support client**

Un service client dédié sera mis en place pour accompagner les utilisateurs à chaque étape de l'organisation. Qu'il s'agisse de conseils sur le choix des prestataires ou d'une assistance technique, notre support sera à disposition pour répondre aux questions. De plus, des consultants spécialisés en organisation d'événements seront référencés afin d'offrir un accompagnement personnalisé.

#### 7. **Option future**

Afin de répondre aux besoins spécifiques des entreprises, un module dédié sera développé pour faciliter l'organisation de séminaires, conférences et autres événements professionnels. Il inclura des outils avancés pour la gestion des inscriptions, la coordination des prestataires et l'optimisation logistique, garantissant ainsi des événements d'affaires bien structurés et réussis.

### 1.5 Conclusion

Au terme de ce premier chapitre, une **analyse approfondie de l'existant** dans le domaine des applications web dédiées aux services événementiels a été réalisée. La présentation de **BigNova**, mon organisme d'accueil, a offert un **cadre concret** et professionnel pour situer mon projet. Cette phase m'a permis de définir clairement les **fonctionnalités attendues** ainsi que les **spécifications techniques** de l'application, en veillant à ce qu'elles répondent aux **besoins réels des utilisateurs**.

En posant ainsi les bases fonctionnelles et techniques du projet, j'ai construit une **vision claire** et **cohérente** de la solution à développer. Cette étude initiale constitue une **étape essentielle** pour orienter les choix à venir.

Le **chapitre suivant** sera consacré à la **présentation de la méthode Scrum**, utilisée pour encadrer et structurer le développement de l'application de manière itérative et collaborative.

## Chapitre 2 Démarche de développement, langage de modélisation et technologies utilisées

### 2.1 Introduction

Ce chapitre est consacré à la présentation de la méthodologie adoptée pour la réalisation de notre projet. Nous y détaillons donc la méthode SCRUM, le langage de modélisation UML, l'architecture MVC ainsi que les technologies utilisées.

### 2.2 La méthode Scrum

Scrum (Schwaber, 2020) est un cadre de gestion de projet agile, utilisé principalement dans le développement logiciel. Il repose sur une approche itérative et collaborative, favorisant l'adaptation rapide aux changements.

Scrum s'appuie sur **trois piliers essentiels** :

- **La transparence**, pour garantir une vision claire et partagée du travail,
- **L'inspection**, pour évaluer régulièrement l'avancement,
- Et **l'adaptation**, pour ajuster les priorités et les actions en fonction des retours et des imprévus.

#### 2.2.1 Rôles de la méthode Scrum

La méthode Scrum repose sur une organisation claire des responsabilités, réparties entre **trois rôles fondamentaux**. Chacun joue un rôle spécifique et complémentaire dans le bon déroulement du projet agile.

##### 2.2.1.1 Le Product Owner (Propriétaire du produit)

Le Product Owner est le représentant des parties prenantes (utilisateurs, clients, direction). Il est responsable de la **vision du produit** et de la **définition des besoins fonctionnels**. Son rôle principal est de gérer le **Product Backlog**, une liste priorisée des fonctionnalités à développer. Il veille à ce que chaque élément du backlog apporte de la valeur métier et reste aligné avec les objectifs stratégiques. Il est également l'interlocuteur privilégié de l'équipe pour clarifier les exigences et répondre aux questions liées au produit.

### 2.2.1.2 Le Scrum Master

Le Scrum Master est le garant de la bonne application de la méthode Scrum. Il agit comme **facilitateur** et **coach** pour l'équipe de développement et le Product Owner. Il veille à ce que l'équipe respecte les principes agiles et les cérémonies Scrum (Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective). Il protège l'équipe des distractions extérieures et **supprime les obstacles** pouvant freiner l'avancement du sprint. Contrairement à un chef de projet traditionnel, le Scrum Master n'est pas hiérarchique mais accompagne l'équipe vers l'autonomie et l'amélioration continue.

### 2.2.1.3 L'équipe de développement (Development Team)

L'équipe de développement est composée de **professionnels pluridisciplinaires** (développeurs, designers, testeurs, etc.) chargés de **concrétiser les éléments du backlog** pendant le sprint. Elle s'organise de manière autonome pour atteindre les objectifs fixés. L'équipe est responsable de la **qualité des livrables** et de la **planification du travail** à l'intérieur du sprint. Elle collabore étroitement avec le Product Owner pour comprendre les besoins et avec le Scrum Master pour améliorer son fonctionnement.

## 2.2.2 Artefacts de Scrum

Les artefacts de Scrum sont des éléments essentiels permettant de **suivre l'avancement du projet** et de **garantir la transparence** entre les membres de l'équipe et les parties prenantes. Ils représentent l'état du travail à chaque étape et servent de support à la planification, à l'inspection et à l'adaptation.

### 2.2.2.1 Le Product Backlog (Carnet de produit)

Le Product Backlog est une **liste priorisée et évolutive** de toutes les fonctionnalités, améliorations, corrections et exigences nécessaires au développement du produit. Il est **géré par le Product Owner**, qui y ajoute, modifie ou réorganise les éléments en fonction des retours, des priorités métiers ou des besoins changeants. Chaque élément du backlog est appelé un **PBI** (Product Backlog Item) et contient une description, une priorité, une estimation et des critères d'acceptation. Ce document évolue tout au long du projet.

### 2.2.2.2 Le Sprint Backlog (Carnet de sprint)

Le Sprint Backlog est un **sous-ensemble du Product Backlog**, sélectionné lors de la planification du sprint (Sprint Planning). Il contient l'ensemble des éléments que l'équipe de développement s'engage à réaliser durant le sprint en cours, ainsi que le **plan de mise en œuvre** associé. Ce backlog est **géré par l'équipe de développement**, qui le met à jour quotidiennement pour suivre l'avancement des tâches. Il permet à l'équipe de rester concentrée sur les objectifs du sprint et facilite la coordination.

### 2.2.2.3 L'incrément de produit (Product Increment)

L'incrément représente le **résultat concret et fonctionnel** du travail effectué à la fin d'un sprint. Il s'agit d'une version potentiellement livrable du produit, qui doit respecter la "**Definition of Done**" (Définition du Terminé) définie par l'équipe. Chaque incrément s'ajoute aux précédents et constitue une nouvelle version plus complète du produit. L'incrément est le **reflet du progrès réalisé**, et il est présenté aux parties prenantes lors de la revue de sprint (Sprint Review).

### 2.2.3 Les événements de la méthode Scrum

La figure 2, représente le cycle de vie de la méthode SCRUM.

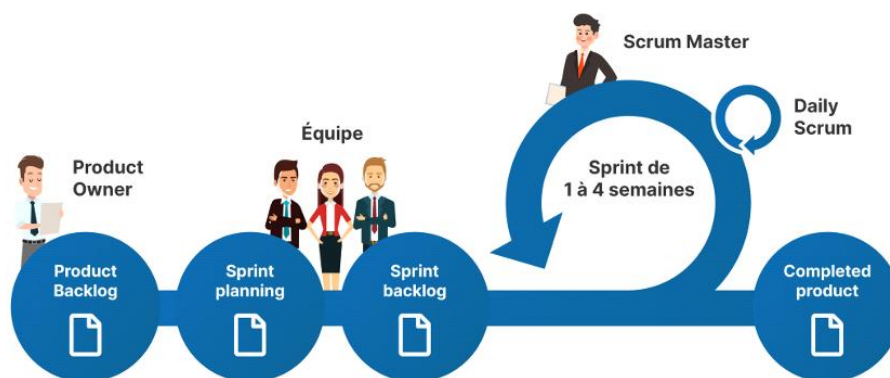


Figure 2: Cycle de vie de la méthode Scrum

La méthode Scrum (Scrum.org. The Home of Scrum. <https://www.scrum.org/>, 2024) Suit plusieurs étapes clés, qui sont :

#### 2.2.3.1 Sprint Planning

La première étape consiste en la planification du sprint où l'équipe sélectionne les tâches à réaliser.

Le **Sprint Planning** est une réunion qui marque le début de chaque sprint. Elle permet à l'équipe Scrum de décider **ce qu'elle va réaliser** au cours de l'itération, et **comment** elle va s'organiser pour y parvenir.

Le **Product Owner** présente les éléments prioritaires du **Product Backlog**, et l'équipe de développement sélectionne ceux qu'elle s'engage à livrer. Ensemble, ils définissent un **objectif de sprint**, qui donne une direction claire au travail à réaliser.

Un **Sprint Backlog** est ensuite constitué, contenant les tâches à accomplir durant le sprint. Cette réunion permet aussi de s'assurer que les exigences sont bien comprises et que tous les critères de qualité sont pris en compte.

En résumé, le Sprint Planning aligne l'équipe sur un **objectif commun** et pose les bases organisationnelles de l'itération.

### 2.2.3.2 Sprint

C'est une période de travail où l'équipe développe les fonctionnalités choisies. Le **sprint** est une itération de durée fixe (généralement de 1 à 4 semaines) durant laquelle l'équipe Scrum développe un **incrément fonctionnel du produit**. Pour qu'un sprint démarre efficacement, certains éléments doivent être en place :

- Un **Sprint Backlog** clair, issu du Product Backlog, contenant les éléments à réaliser.
- Un **objectif de sprint**, défini lors de la planification, servant de fil conducteur tout au long de l'itération.
- Une **Définition de Terminé** partagée, qui garantit la qualité et la complétude du travail.
- Les **ressources techniques** nécessaires (environnements, outils, accès, etc.).
- La **capacité réelle de l'équipe**, tenant compte de la disponibilité et des compétences.

À la fin du sprint, l'équipe livre un **incrément du produit**, c'est-à-dire une version potentiellement livrable contenant les éléments complétés et validés selon les critères définis.

Le sprint permet aussi de recueillir des **retours (feedbacks)** lors de la revue, et d'identifier des pistes d'amélioration pendant la rétrospective. Ces enseignements nourrissent l'amélioration continue.

Enfin, le **Product Backlog est mis à jour** par le Product Owner en fonction des retours, des priorités révisées ou des découvertes techniques.

### 2.2.3.3 Daily Scrum

C'est une réunion quotidienne rapide (15 min) pour suivre l'avancement et ajuster les actions. Le **Daily Scrum**, ou réunion quotidienne, est un rituel court (15 minutes) qui a lieu chaque jour du sprint. Il permet à l'équipe de développement de **synchroniser ses actions**, de **faire le point sur l'avancement**, et de **détecter rapidement les obstacles**.

Pour être efficace, cette réunion s'appuie sur plusieurs éléments :

- Le **Sprint Backlog**, qui sert de référence pour suivre l'état des tâches à réaliser.
  - **L'avancement depuis la veille**, partagé par chaque membre, permettant une vision claire du progrès collectif.
  - **L'objectif du sprint**, qui reste en toile de fond pour orienter les échanges et recentrer les efforts si besoin.
  - **Les blocages identifiés**, qui sont signalés pour être résolus rapidement et ne pas freiner l'équipe.
  - Et enfin, **le plan d'action de chacun jusqu'au prochain Daily**, qui favorise l'engagement, la coordination et la réactivité.

Le Daily Scrum permet ainsi de maintenir un rythme constant, de fluidifier la communication et d'assurer une progression cohérente vers l'objectif du sprint.

#### 2.2.3.4 Sprint Review

La **Sprint Review** est une réunion tenue à la fin de chaque sprint pour **présenter le travail réalisé** aux parties prenantes et recueillir leurs **retours**. Elle permet d'évaluer les résultats obtenus, d'ajuster le Product Backlog, et de définir les prochaines priorités.

Parmi les éléments mobilisés, on retrouve en premier lieu **l'incrément du produit**, qui correspond aux fonctionnalités terminées, conformes à la définition de Done, et potentiellement livrables. **L'objectif du sprint** sert de référence pour évaluer si les engagements ont été atteints. Le **Sprint Backlog** permet de visualiser ce qui a été complété ou non, et d'expliquer les écarts éventuels.

Les **feedbacks des parties prenantes** sont essentiels : ils enrichissent la vision du produit, révèlent les attentes non couvertes et orientent les futures évolutions. Enfin, les **problèmes rencontrés** et quelques **données de performance** (comme la vélocité) aident à évaluer l'efficacité globale du sprint.

À l'issue de la Sprint Review, plusieurs résultats concrets en découlent :

- La **validation de l'incrément**, si les fonctionnalités respectent les critères définis ;
- La **mise à jour du Product Backlog**, avec ajout, suppression ou révision d'éléments ;
- Et une **vision partagée des prochaines étapes**, permettant d'aligner les priorités avec les besoins métier.

La Sprint Review joue ainsi un rôle clé dans la **transparence**, **l'adaptation** et la **construction continue** du produit avec l'ensemble des acteurs concernés.

### 2.2.3.5 Sprint Retrospective

Analyse de ce qui a bien fonctionné et des améliorations à apporter pour le prochain sprint. La **Sprint Retrospective** est une réunion de réflexion interne à l'équipe Scrum, organisée à la fin de chaque sprint. Elle vise à analyser **le déroulement de l'itération écoulée** pour en tirer des **enseignements utiles et applicables**.

Parmi les éléments pris en compte, on retrouve :

- Le **sprint passé**, avec l'ensemble des actions, événements et interactions.
- Les **feedbacks de l'équipe** (Scrum Master, Product Owner, développeurs), qui permettent d'identifier les points positifs, les problèmes rencontrés et les pistes d'amélioration.
- L'**objectif du sprint**, qui sert à évaluer l'écart entre les prévisions et les résultats obtenus.
- Les **blocages techniques, organisationnels ou relationnels**, qui doivent être compris pour être évités à l'avenir.
- Les **pratiques et outils utilisés**, afin d'évaluer leur efficacité.
- Et enfin, des **données factuelles** comme la vélocité ou le taux de réalisation, pour appuyer les constats.

À l'issue de la réunion, l'équipe élabore une **liste d'améliorations à mettre en œuvre**. Cela peut concerner l'organisation, les outils, les méthodes de travail ou la communication. Un **plan d'action clair** est établi pour le prochain sprint, accompagné d'un **engagement collectif** à appliquer les changements décidés.

Un **suivi des actions précédentes** est aussi réalisé pour mesurer les effets des améliorations déjà testées. Ce processus régulier d'analyse et d'adaptation permet à l'équipe de **progresser continuellement** tout au long du projet.

## 2.3 Langage de modélisation

Pour concevoir mon application de manière structurée, nous avons choisi d'utiliser UML (Unified Modeling Language), un langage de modélisation graphique standardisé. Il permet de représenter à la fois la structure statique du système (données, relations) et son comportement dynamique (interactions, processus). Les principaux diagrammes utilisés sont : le diagramme de cas d'utilisation pour illustrer les interactions entre les acteurs et le système, le diagramme de classes pour modéliser les entités et leurs relations, le diagramme de séquence pour représenter les échanges entre objets dans le temps. UML facilite ainsi la compréhension, la conception et la communication autour du système, et servira de base à la modélisation dans le chapitre suivant.

## 2.4 Technologies utilisées

### 2.4.1 Front-end

#### 2.4.1.1 HTML

**HTML** (*HyperText Markup Language*) est un **langage de balisage** utilisé pour structurer le contenu des pages web (Duckett, 2011) (Mozilla Developer Network (MDN). . HTML et CSS documentation. <https://developer.mozilla.org/fr/>, 2024). Il permet d'organiser les éléments comme les **titres**, **paragraphes**, **images**, **liens** ou encore les **tableaux**, à l'aide de **balises** spécifiques.

HTML fonctionne en complémentarité avec **CSS** pour la mise en forme et **JavaScript** pour l'interactivité. Avec l'arrivée de **HTML5**, de nouvelles **balises sémantiques** telles que `<header>`, `<section>`, ou `<article>` ont été introduites, améliorant ainsi la **lisibilité du code**, l'**accessibilité** et le **référencement naturel (SEO)**.

Interprété par les navigateurs, HTML est à la base de **toute page web**. Il constitue un **standard incontournable** du développement web, permettant d'assurer une structure claire, logique et compatible sur tous les supports.

#### 2.4.1.2 CSS

Le **CSS** (Cascading Style Sheets) est un **langage de style** qui permet de définir l'apparence des éléments HTML d'une page web (W3Schools., 2024). Il contrôle divers aspects visuels tels que les **couleurs**, les **polices**, les **marges**, ou encore la **mise en page**, rendant les sites plus **esthétiques**, **ergonomiques** et **responsifs**.

Les styles peuvent être définis de plusieurs manières, mais l'utilisation de **fichiers externes (.css)** est courante car elle facilite la **maintenance**, la **réutilisation du code** et la **séparation des préoccupations** entre structure (HTML) et présentation (CSS).

Avec **CSS3**, des fonctionnalités avancées comme les **animations**, les **transitions**, **Flexbox** et **CSS Grid** permettent de concevoir des interfaces modernes, fluides et dynamiques. Outil indispensable du développement web, CSS joue un rôle essentiel dans l'**expérience utilisateur**, l'**accessibilité** et l'**attractivité visuelle** des sites internet.

#### 2.4.1.3 JS

JS est un langage de programmation interprété, orienté objets, principalement utilisé pour le développement web. Il permet de manipuler le DOM, de gérer les événements et d'ajouter des comportements dynamiques aux pages HTML

C'est un excellent choix car il est universellement supporté par tous les navigateurs, dispose d'une vaste communauté, et permet de développer à la fois côté client et côté serveur (avec Node.js), assurant ainsi une grande cohérence dans les projets web.

#### 2.4.1.4 NextJS

Next.js est un framework open-source basé sur React qui facilite la création d'applications web performantes. Il prend en charge le rendu côté serveur (SSR), la génération de sites statiques (SSG) et le routage automatique.

Ce framework est un choix pertinent car il améliore considérablement le référencement (SEO), la rapidité de chargement des pages et la maintenabilité du code. Il offre une structure claire, une optimisation automatique, et une expérience développeur moderne et productive.

#### 2.4.1.5 React

**React** est une bibliothèque JavaScript développée par **Facebook**, conçue pour créer des **interfaces utilisateur dynamiques et réactives** (Grinberg, 2018) (Media., 2020). Elle repose sur une **architecture basée sur des composants**, ce qui permet de concevoir des éléments **modulaires** et **réutilisables**, facilitant ainsi la maintenance et l'évolution des interfaces.

L'un de ses principaux atouts est le **Virtual DOM**, qui permet d'optimiser les mises à jour de l'interface en ne modifiant que les éléments nécessaires, améliorant ainsi les **performances**. React utilise un **flux de données unidirectionnel**, rendant les applications plus **prévisibles** et **faciles à déboguer**.

Grâce aux **React Hooks**, la gestion de l'état est simplifiée et plus intuitive. Enfin, React s'intègre parfaitement avec d'autres outils comme **Redux** pour la gestion d'état globale, **Next.js** pour le rendu côté serveur, ou encore **React Native** pour le développement mobile. Ces caractéristiques font de React un **outil incontournable** pour le développement d'applications web modernes et performantes.

### 2.4.2 Back-end

#### 2.4.2.1 NodeJS

**Node.js** est un environnement d'exécution JavaScript côté serveur, introduit par Cantelon en 2014. Il repose sur le moteur **V8** de Google Chrome, ce qui lui confère de hautes performances. Conçu pour être **asynchrone**, **non bloquant** et **orienté événements**, il est particulièrement adapté aux applications en temps réel telles que les **API**, les **services de chat** ou de **streaming**.

Node.js est couramment associé à **Express.js**, un framework minimaliste facilitant la création d'applications web. Il s'appuie sur **NPM** (Node Package Manager), un gestionnaire de paquets très riche, qui

fournit un vaste écosystème de bibliothèques. Cet environnement permet un développement **fullstack JavaScript**, assurant ainsi une cohérence entre le frontend et le backend. Grâce à sa **scalabilité** et ses **performances élevées**, Node.js est aujourd'hui un choix privilégié pour le développement d'applications web modernes.

### 2.4.2.2 ExpressJS

**Express.js** est un framework web minimaliste et flexible pour **Node.js**, présenté par Holtz en 2021. Il facilite la création d'**API RESTful** et d'**applications web** en simplifiant la gestion des **routes**, des **requêtes HTTP** et des **middlewares**. Grâce à son système de middleware, il est facile d'ajouter des fonctionnalités comme l'**authentification**, la **journalisation** ou la **gestion des erreurs**.

Express.js s'intègre aisément avec des bases de données comme **MongoDB** ou des bibliothèques temps réel telles que **Socket.io**. Sa simplicité, sa légèreté et sa rapidité en font un outil de choix pour développer des applications **modernes, scalables** et **maintenables** avec Node.js.

### 2.4.2.3 MongoDB

**MongoDB** est une base de données **NoSQL orientée documents**, développée par MongoDB Inc. (2024). Elle stocke les données au format **JSON** ou **BSON**, ce qui lui confère une **grande flexibilité** par rapport aux bases relationnelles traditionnelles. MongoDB permet de gérer des données **non structurées** sans schéma fixe, et offre une **scalabilité horizontale** efficace grâce au **sharding** (partitionnement des données).

Souvent intégrée dans les architectures **MEAN** ou **MERN** (avec Node.js et Express.js), MongoDB est particulièrement adaptée aux **applications web et mobiles**, ainsi qu'aux domaines du **Big Data** et de l'**IoT**. Elle se distingue par sa **rapidité**, sa **performance**, et sa capacité à gérer de **grands volumes de données** de manière fluide et évolutive.

## 2.5 Conclusion

Ce chapitre s'est concentré sur la présentation de la **methodologie Scrum**, choisie pour encadrer le développement de mon projet. À ce stade, il ne s'agit pas encore de modélisation ou de conception technique, mais d'une **mise en place du cadre méthodologique** qui guidera toutes les étapes futures. En décrivant les rôles, les artefacts et les événements clés de Scrum, j'ai défini une approche de travail itérative et collaborative, adaptée aux besoins de mon application.

La **modélisation du projet**, à travers notamment l'utilisation d'UML, sera abordée dans le **chapitre suivant**, où je poserai les bases concrètes de la conception du système.

## Chapitre 3 Réalisation du projet selon la démarche SCRUM

### Introduction

Dans le cadre de ce projet de gestion d'événements, une approche astucieuse a été adoptée afin de garantir un développement progressif, flexible et orienté utilisateur. La méthode **Scrum**, particulièrement adaptée aux projets dynamiques, a été choisie pour organiser les différentes étapes de réalisation du système.

Ce chapitre décrit la mise en œuvre concrète de cette démarche. Il débute par l'**identification des acteurs** du système, suivie de la **répartition des rôles Scrum** (Product Owner, Scrum Master, équipe de développement) et la définition des **user stories**, qui traduisent les besoins des utilisateurs en fonctionnalités claires.

Chaque **sprint** est ensuite présenté en détail, avec les **cas d'utilisation** associés, dont les scénarios sont décrits **sous forme de tableaux** textuels pour plus de clarté. Ces cas sont accompagnés de **diagrammes de séquence**, de **diagrammes de classes**, ainsi que des **interfaces graphiques** réalisées pour chaque fonctionnalité. Cette structure permet de suivre l'évolution progressive du projet vers un produit final cohérent, fonctionnel et adapté aux besoins réels des utilisateurs.

### 3.1 Identification des acteurs

#### 3.1.1 Administrateur

L'administrateur occupe un rôle central dans la gestion des fêtes et événements. Il est responsable de la création, de la modification et de l'annulation des événements, tout en assurant la planification des ressources nécessaires (salles, équipements, prestataires). Il gère également les utilisateurs en attribuant des rôles et en suivant les inscriptions.

Sur le plan financier, l'administrateur supervise les paiements et assure le suivi budgétaire. Il joue aussi un rôle dans la communication en envoyant des invitations, en promouvant les événements et en assurant la visibilité de la plateforme. En parallèle, il veille au respect des réglementations, à la sécurité des événements et à la gestion des éventuels incidents.

En plus de ses nombreuses responsabilités, l'administrateur peut consulter le site pour accéder aux événements en cours, aux services proposés et aux informations actualisées. Son rôle est essentiel pour garantir le bon déroulement, la conformité et la réussite de chaque événement organisé sur la plateforme.

### 3.1.2 Partenaire

Le partenaire est un prestataire essentiel à la réussite des événements. Il propose une gamme variée de services tels que la restauration, la décoration, l'animation ou encore la location de matériel. Il définit ses offres, fixe ses tarifs et gère les réservations en collaboration avec l'administrateur.

Le partenaire assure la livraison et l'installation de ses prestations, tout en s'adaptant aux besoins spécifiques et aux éventuelles modifications. Il prend en charge la facturation, le suivi des paiements et veille à l'optimisation continue de ses services. Soucieux de la satisfaction des clients, il garantit la qualité de ses prestations et traite les réclamations éventuelles.

En complément de ses responsabilités, le partenaire peut accéder à la plateforme pour consulter les événements, suivre ses services proposés et rester informé des dernières mises à jour. Son rôle contribue directement à la qualité et à la fluidité de l'organisation des événements.

### 3.1.3 Visiteur anonyme

Le visiteur anonyme occupe un rôle passif mais important dans la plateforme. Sans avoir besoin de créer un compte, il peut accéder librement au site et consulter les différentes offres d'événements et de services proposés. Il a la possibilité de visualiser les détails des événements (date, lieu, description, prestations associées), ainsi que les services disponibles comme la restauration, l'animation ou la location de matériel.

Cependant, le visiteur ne peut ni effectuer de réservation ni interagir avec les prestataires sans s'inscrire. Son rôle consiste principalement à **découvrir l'offre** et à **se familiariser avec la plateforme**, dans la perspective de devenir un futur **utilisateur inscrit**.

### 3.1.4 Client

Le client est un utilisateur inscrit qui dispose d'un accès complet aux fonctionnalités de réservation de la plateforme. Il peut consulter le **catalogue des événements**, sélectionner ceux qui l'intéressent et effectuer des **réservations** en quelques clics. Il a également la possibilité de personnaliser son événement en choisissant des **services spécifiques** proposés par les partenaires, tels qu'un traiteur, une animation ou une décoration.

Le client gère ses réservations de manière autonome, suit l'évolution de ses **paiements**, et peut **modifier** ou **annuler** ses choix en fonction des conditions prévues. En complément, il peut naviguer librement sur le site pour consulter les événements, les services disponibles et les informations régulièrement mises à jour.

### 3.2 Répartition des rôles

Dans le tableau 1, la présentation des personnes assignées aux rôles suivant la méthode Scrum.

**Tableau 1 : Répartition des rôles**

Rôles Scrum	Personnes assignées
Product Owner	Mr Bekka Reda
Scrum Master	Mme Nouicer Amina
Development Team	Mr Sellami Abderrahim

### 3.3 User stories

Une « user story » est une courte phrase qui décrit ce que veut faire un utilisateur avec le produit, et pourquoi. Elle sert à guider le développement d'une fonctionnalité.

Le tableau 2 illustre le User stories pour mon projet.

*Tableau 2 : User stories*

Acteur	Stories
<b>Administrateur</b>	<p>L'administrateur peut :</p> <p>S'inscrire</p> <p>S'authentifier</p> <p>Gérer les clients :</p> <ul style="list-style-type: none"> <li>• Ajouter un client</li> <li>• Modifier un client</li> <li>• Supprimer un client</li> </ul> <p>Gérer les partenaires :</p> <ul style="list-style-type: none"> <li>• Ajouter un partenaire</li> <li>• Modifier un partenaire</li> <li>• Supprimer un partenaire</li> </ul> <p>Gérer les événements :</p>

	<ul style="list-style-type: none"> <li>• Ajouter un événement</li> <li>• Modifier un événement</li> <li>• Supprimer un événement</li> </ul>
<b>Partenaire</b>	<p>Le partenaire peut :</p> <p>S'inscrire</p> <p>S'authentifier</p> <p>Gérer les services</p> <ul style="list-style-type: none"> <li>• Ajouter un service</li> <li>• Modifier un service</li> <li>• Supprimer un service</li> </ul>
<b>Visiteur anonyme</b>	<p>Le visiteur anonyme peut :</p> <p>Consulter le site</p>
<b>Client</b>	<p>Le client peut :</p> <p>S'inscrire</p> <p>S'authentifier</p> <p>Gérer les réservations :</p> <ul style="list-style-type: none"> <li>• Ajouter une réservation</li> <li>• Modifier une réservation</li> <li>• Supprimer une réservation</li> </ul> <p>Consulter le site</p>

### 3.4 Diagramme de contexte

La figure 3 représente le lien entre le système de gestion de fêtes et événements et les acteurs agissant dans l'application.

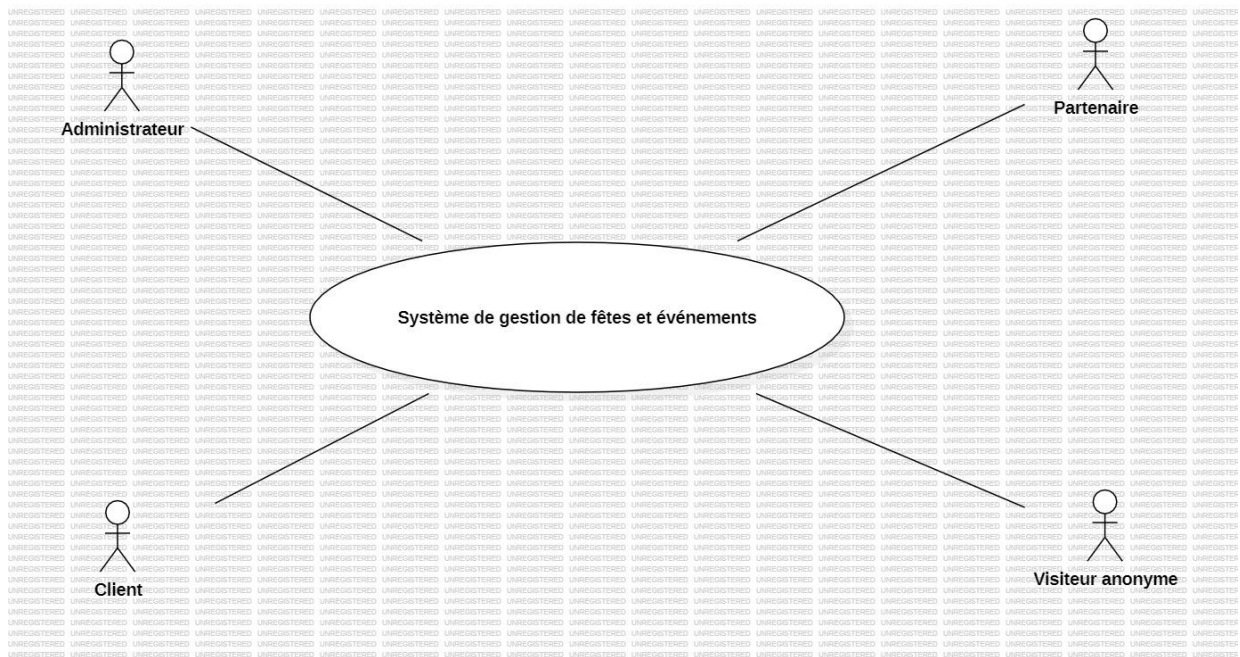
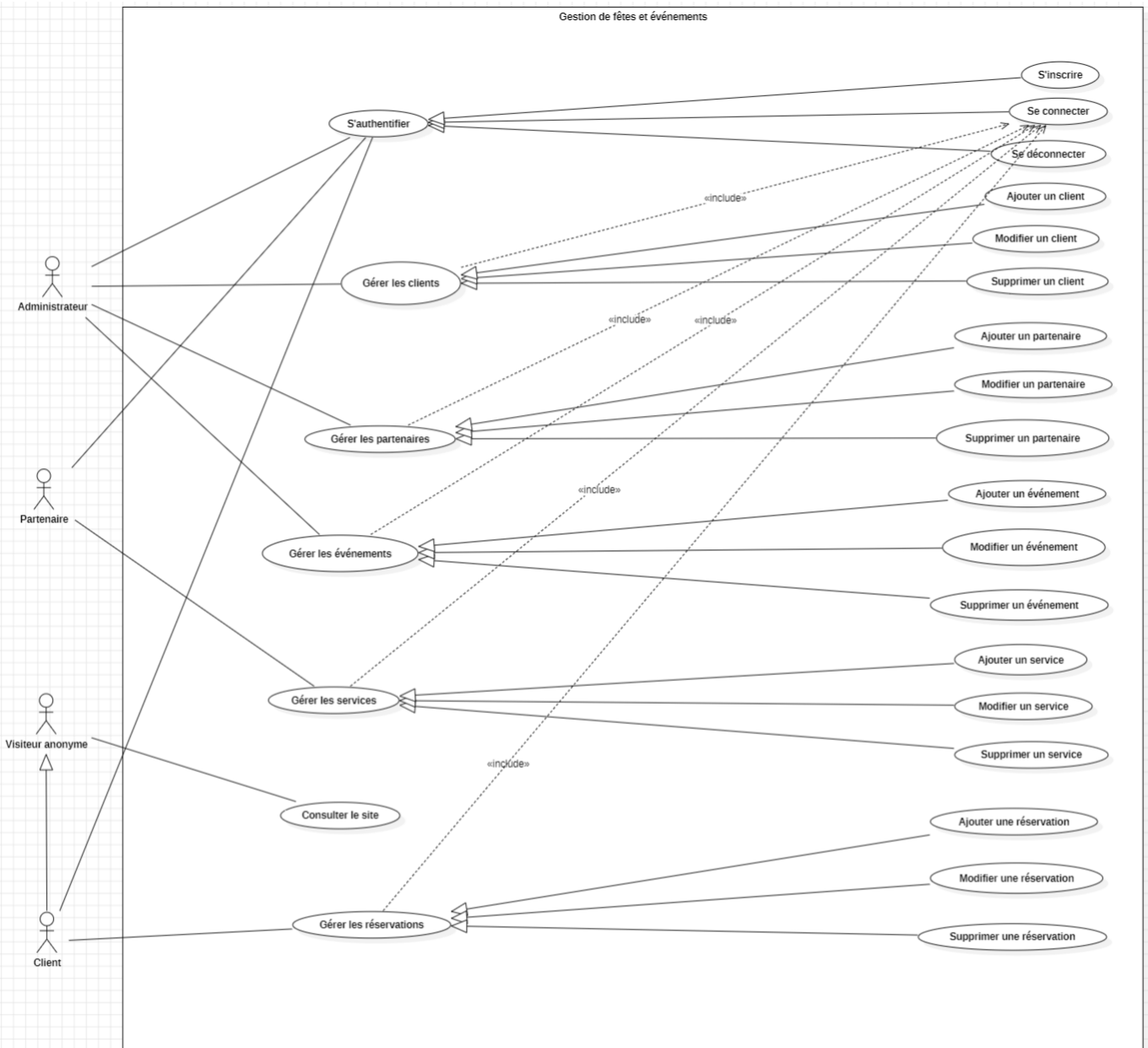


Figure 3: Diagramme de contexte

### 3.5 Diagramme de cas d'utilisation global

La figure 4 illustre le diagramme de cas d'utilisation global de notre future application. Ce diagramme démontre les principales interactions entre les acteurs et notre application et fournit une vue générale des relations et des dépendances entre les différents cas d'utilisation.



**Figure 4: Diagramme de cas d'utilisation global**

### 3.6 Product Backlog

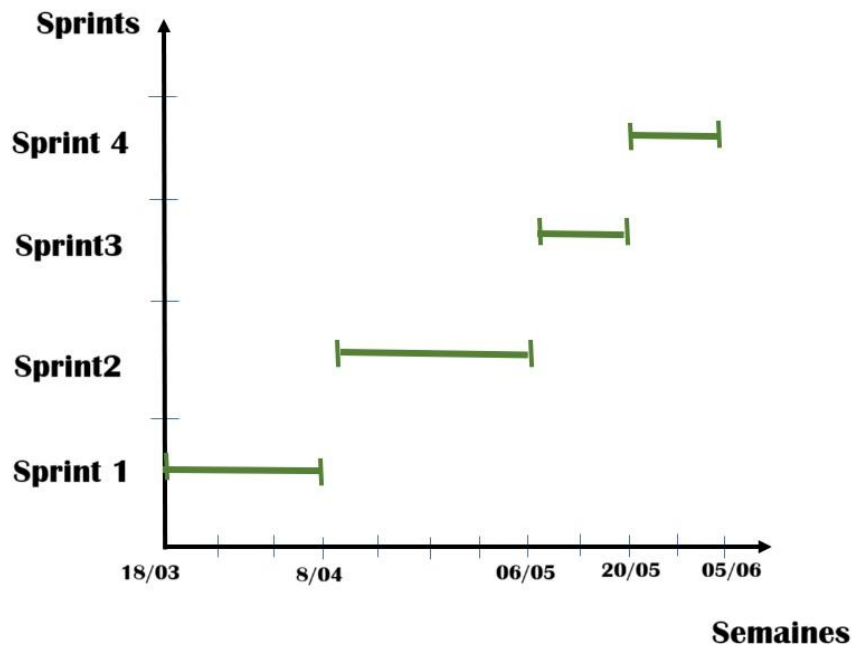
Le Product backlog (carnet de produit) qui contient toutes les fonctionnalités à développer priorisées par le product owner. Le tableau 3 suivant récapitule les différentes fonctionnalités de notre future application regroupées en sprints en fonction de leur priorité associée.

**Tableau 3 : Product Backlog**

<b>Sprint</b>	<b>Items</b>	<b>En tant que</b>	<b>Je veux</b>	<b>Priorité</b>
<b>1</b>	S'inscrire S'authentifier	Administrateur, Partenaire, Client	M'inscrire M'authentifier	1
<b>2</b>	A. Gérer les clients B. Gérer les partenaires C. Gérer les services	A. Administrateur B. Administrateur C. Partenaire	A. Ajouter un client Modifier un client Supprimer un client B. Ajouter un partenaire Modifier un partenaire Supprimer un partenaire C. Ajouter un service Modifier un service Supprimer un service	2
<b>3</b>	Gérer les réservations	Client	Ajouter une réservation Modifier une réservation Supprimer une réservation	3
<b>4</b>	Gérer les événements	Administrateur	Ajouter un événement Modifier un événement Supprimer un événement	4

### **3.7 Planification des sprints**

Après avoir identifié et défini les fonctionnalités à développer, celles-ci ont été réparties en plusieurs sprints selon leur priorité. Chaque sprint regroupe un ensemble cohérent de fonctionnalités à réaliser dans un délai déterminé. L'étape suivante consiste alors à établir une planification précisant la durée de réalisation de chaque sprint, ainsi que ses dates de début et de fin, comme illustré par le diagramme de Gantt suivant.



**Figure 5 : Diagramme de Gantt**

### **3.8 Sprint 1**

Ce sprint dure trois semaines, il comprend les tâches suivantes :

- S'inscrire
- S'authentifier

Le résultat de ce sprint est un incrément livrable représentant la première partie fonctionnelle de notre application. Dans ce qui suit, nous présentons le diagramme de cas d'utilisation correspondant, les descriptions textuelles des cas d'utilisation, les diagrammes de séquence, ainsi que le diagramme de classes participantes.

Enfin, nous illustrons les interfaces utilisateur développées au cours de ce sprint à l'aide de captures d'écran.

#### **3.8.1 Descriptions textuelles**

Les tableaux 4 et 5 représentent les descriptions textuelles du sprint 1.

**Tableau 4 : Description textuelle du cas d'utilisation S'inscrire**

<b>Titre</b>	S'inscrire
<b>Acteurs</b>	Administrateur, Partenaire, Client
<b>Objectifs</b>	Faire une inscription pour avoir un compte qui sera utilisé pour se connecter
<b>Pré conditions</b>	/
<b>Scénario nominal</b>	1/ Accéder à l'interface d'inscription 2/ Remplir les champs du formulaire d'inscription 3/ Cliquer sur le bouton S'inscrire 4/ Effectuer un traitement pour vérifier la syntaxe des données 5/ Effectuer un traitement pour vérifier si le compte existe déjà dans la BDD 6/ Sauvegarder le nouveau compte dans la BDD 7/ Nouveau compte créé avec succès
<b>Scénario alternatif</b>	5/ Erreur dans la syntaxe des données, réessayer 6/ Erreur Le compte existe déjà
<b>Post conditions</b>	Compte créé avec succès

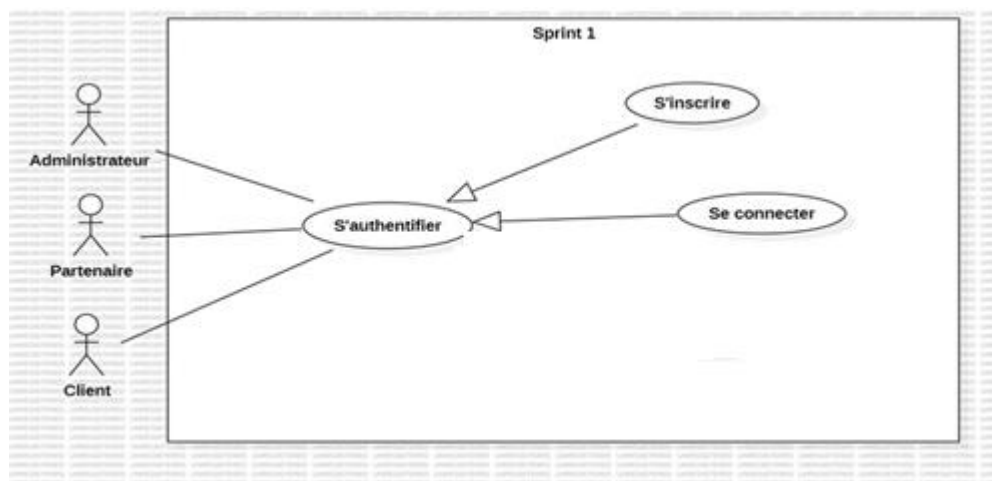
**Tableau 5 : Description textuelle du cas d'utilisation S'authentifier**

<b>Titre</b>	S'authentifier
<b>Acteurs</b>	Administrateur, Partenaire, Client
<b>Objectifs</b>	Permettre à l'utilisateur de s'authentifier grâce à un compte
<b>Pré conditions</b>	L'utilisateur doit avoir un compte
<b>Scénario nominal</b>	1/ L'utilisateur accède à la page d'authentification 2/ L'utilisateur remplit les champs du formulaire de connexion 3/ L'utilisateur clique sur le bouton S'authentifier

	<p>4/ Effectuer un traitement pour vérifier la syntaxe des données</p> <p>5/ Effectuer un traitement pour rechercher le compte au niveau de la BDD</p> <p>6/ Utilisateur authentifié avec succès</p>
<b>Scénario alternatif</b>	<p>5/ Erreur dans la syntaxe des données, réessayer</p> <p>6/ Erreur lors de l'authentification, email ou mot de passe incorrect</p>
<b>Post conditions</b>	Utilisateur authentifié avec succès

### 3.8.2 Diagramme de cas d'utilisation du sprint 1

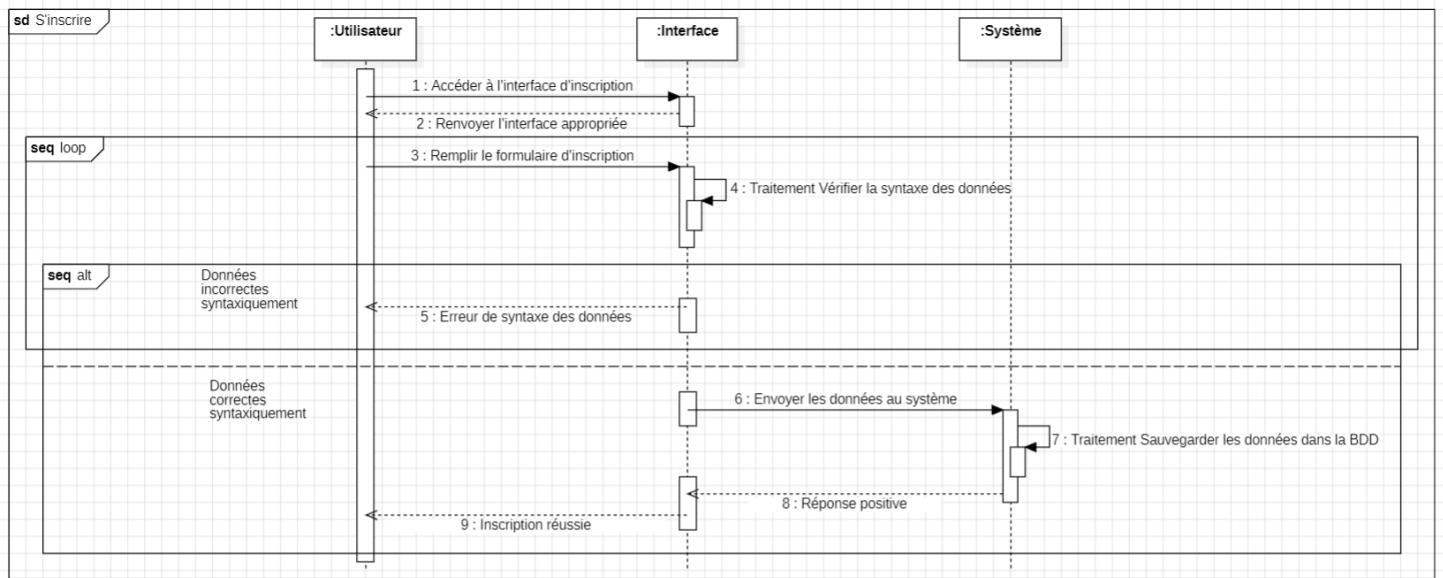
Dans la figure 6, on regroupe les informations des trois tableaux qui décrivent les cas d'utilisation du sprint 1.



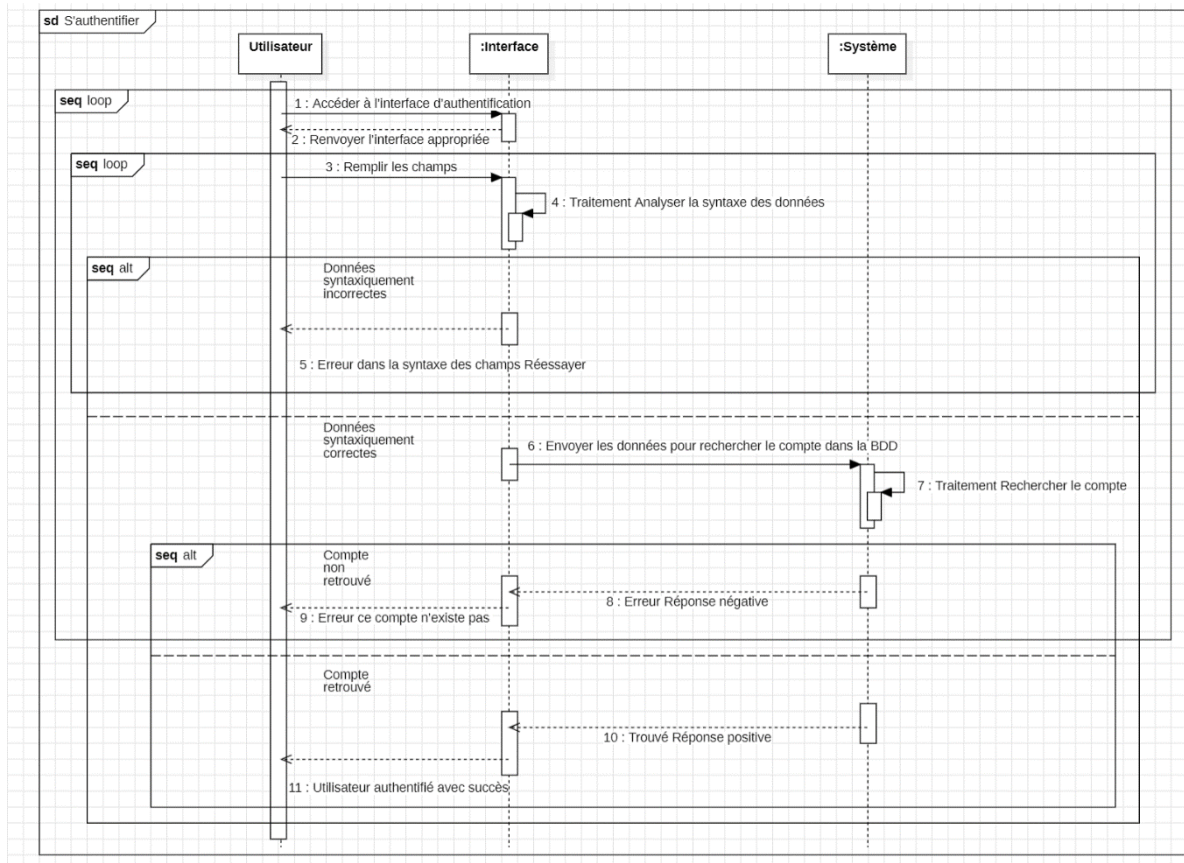
**Figure 6 : Diagramme de cas d'utilisation du sprint 1**

### 3.8.3 Diagrammes de séquence

Les figures 7 et 8 représentent respectivement les diagrammes de séquence des cas d'utilisation s'inscrire et s'authentifier.



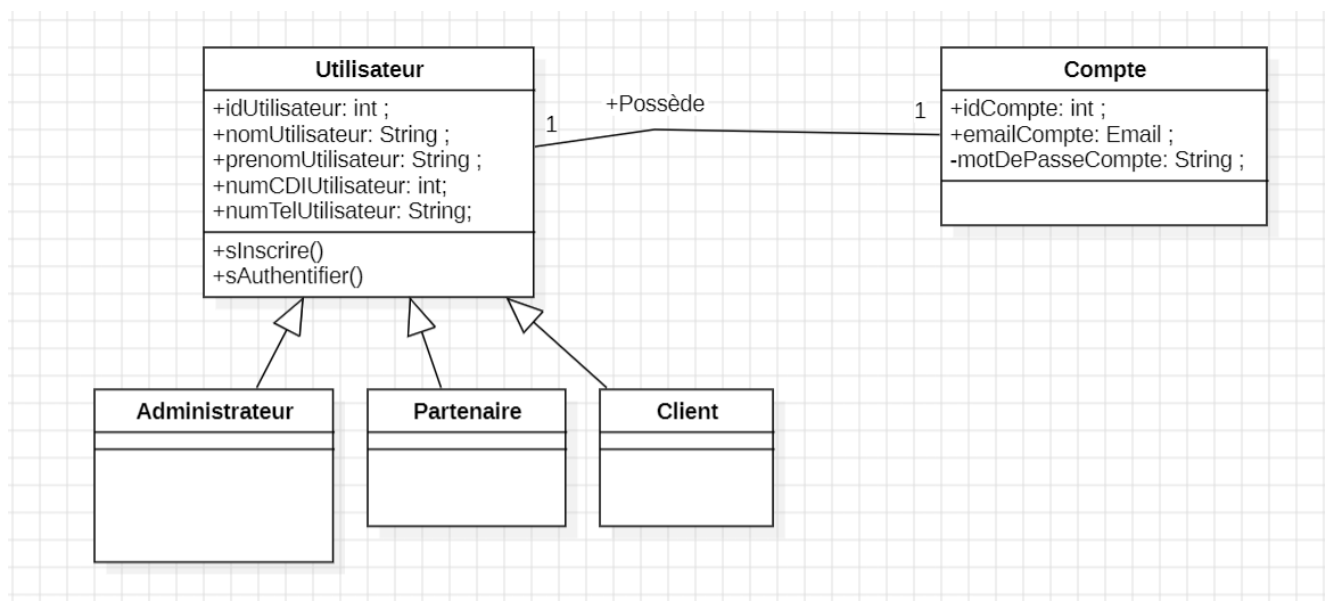
**Figure 7: Diagramme de séquence du cas d'utilisation S'inscrire**



**Figure 8: Diagramme de séquence du cas d'utilisation S'authentifier**

### 3.8.4 Diagramme de classes participantes

La figure 9 représente le diagramme de classes participantes du sprint 1.



**Figure 9 : Diagramme de classes participantes du sprint 1**

### 3.8.5 Interfaces utilisateur

Nous commençons par l'interface utilisateur de la page d'accueil de notre application.

La figure suivante représente l'interface d'accueil pour le client

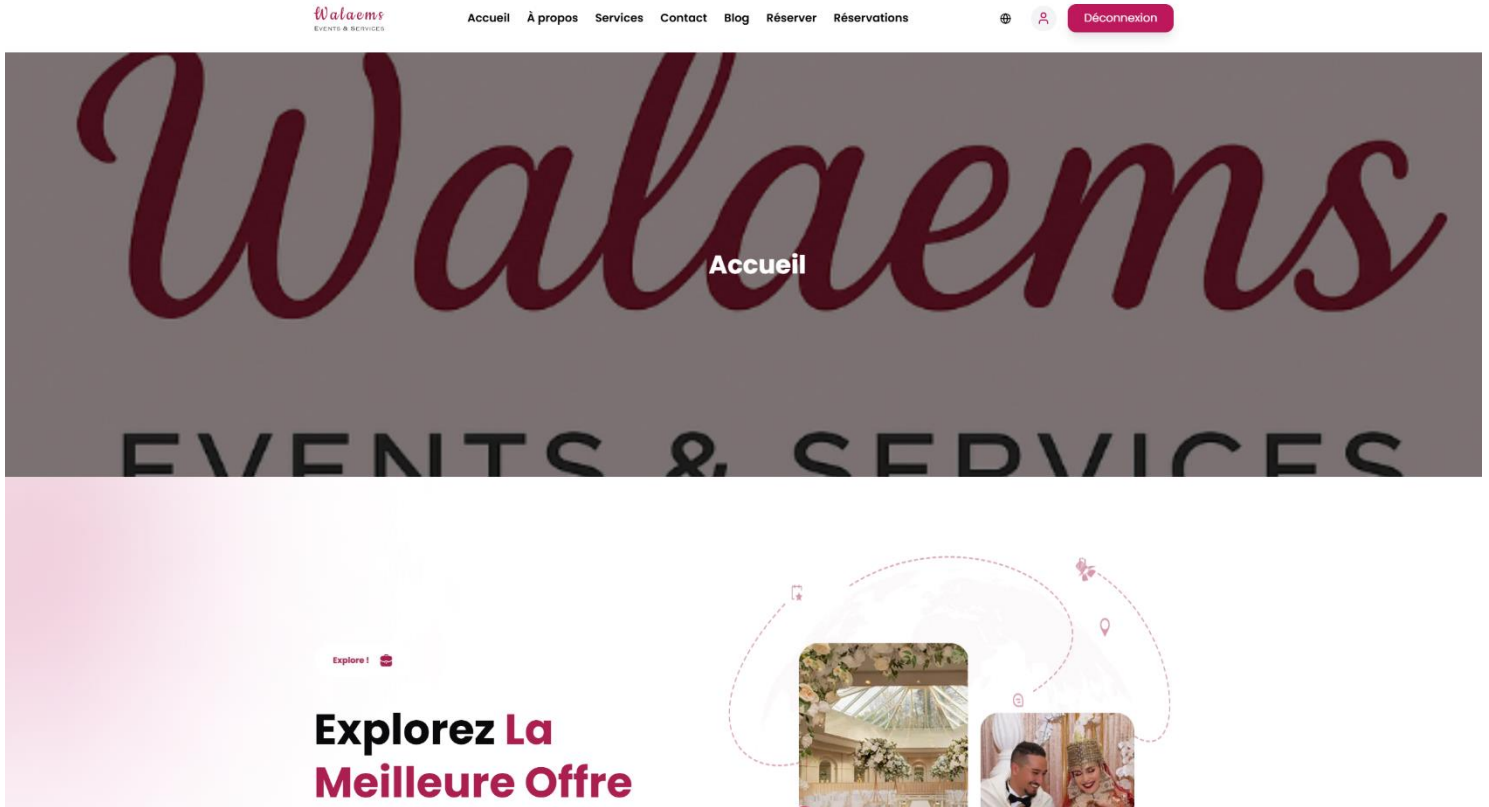


Figure 10: Interface d'accueil Client

Après sa connexion, l'administrateur accède à cette interface, qui constitue sa page d'accueil. Elle reflète les évolutions et les mises à jour récentes apportées à la plateforme.



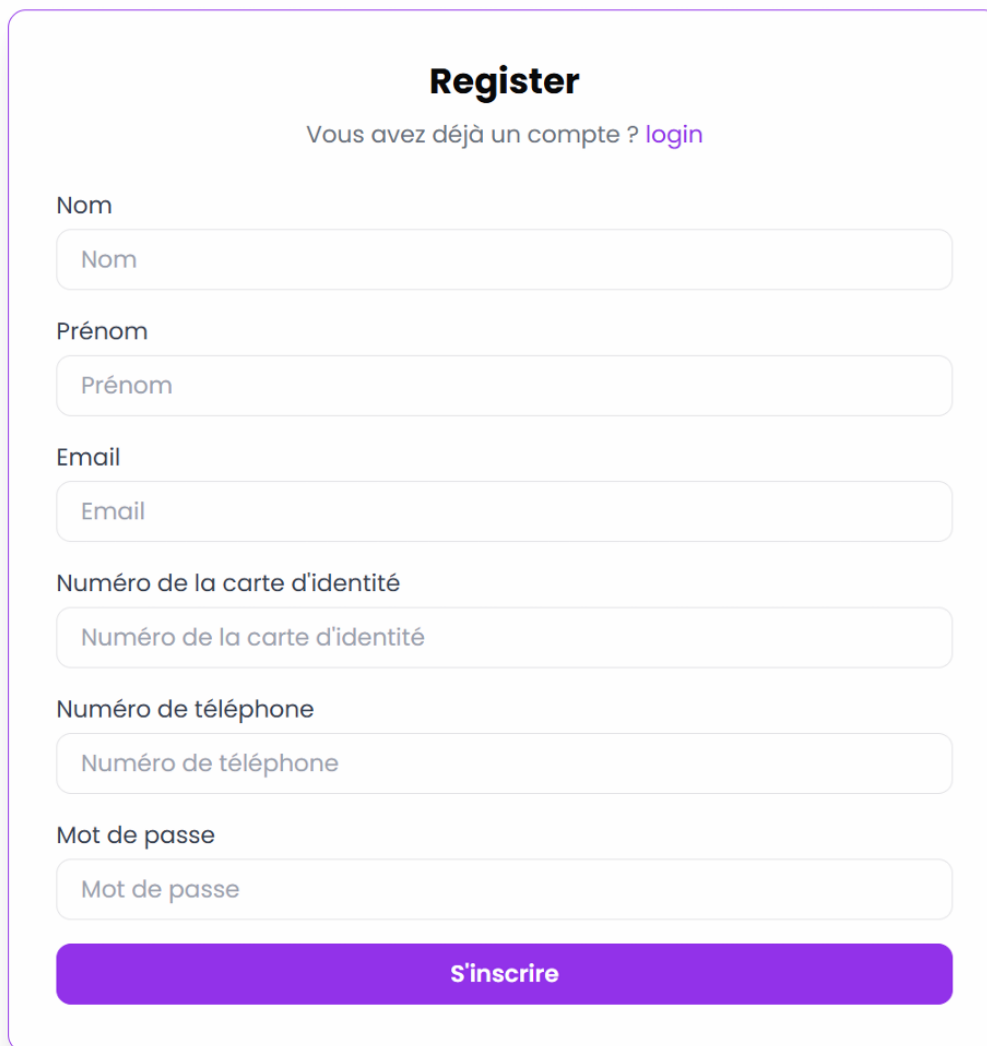
Figure 11: Interface d'accueil Administrateur

La figure suivante représente l'interface d'accueil pour le partenaire



**Figure 12: Interface d'accueil Partenaire**

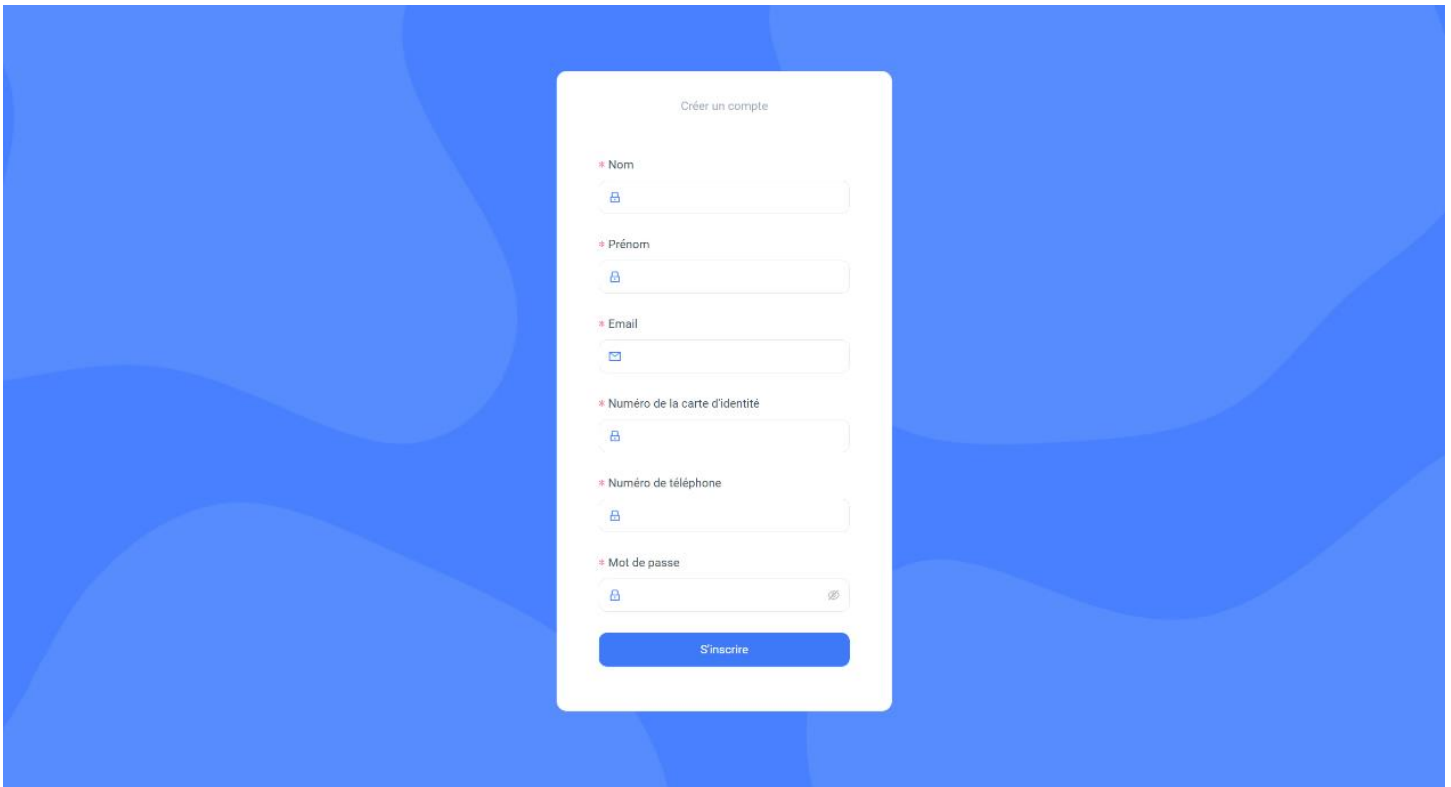
La figure suivante illustre le formulaire d'inscription d'un nouvel **utilisateur** à notre plateforme d'événements et fêtes.



The image shows a registration form titled "Register". At the top, it asks "Vous avez déjà un compte ?" followed by a purple link "login". Below this are seven input fields, each with a label and a placeholder text: "Nom" (Nom), "Prénom" (Prénom), "Email" (Email), "Numéro de la carte d'identité" (Numéro de la carte d'identité), "Numéro de téléphone" (Numéro de téléphone), and "Mot de passe" (Mot de passe). At the bottom of the form is a large purple button with the text "S'inscrire" in white.

**Figure 13: Interface S'inscrire pour le client**

La figure suivante représente l'interface d'inscription pour le partenaire et l'administrateur.

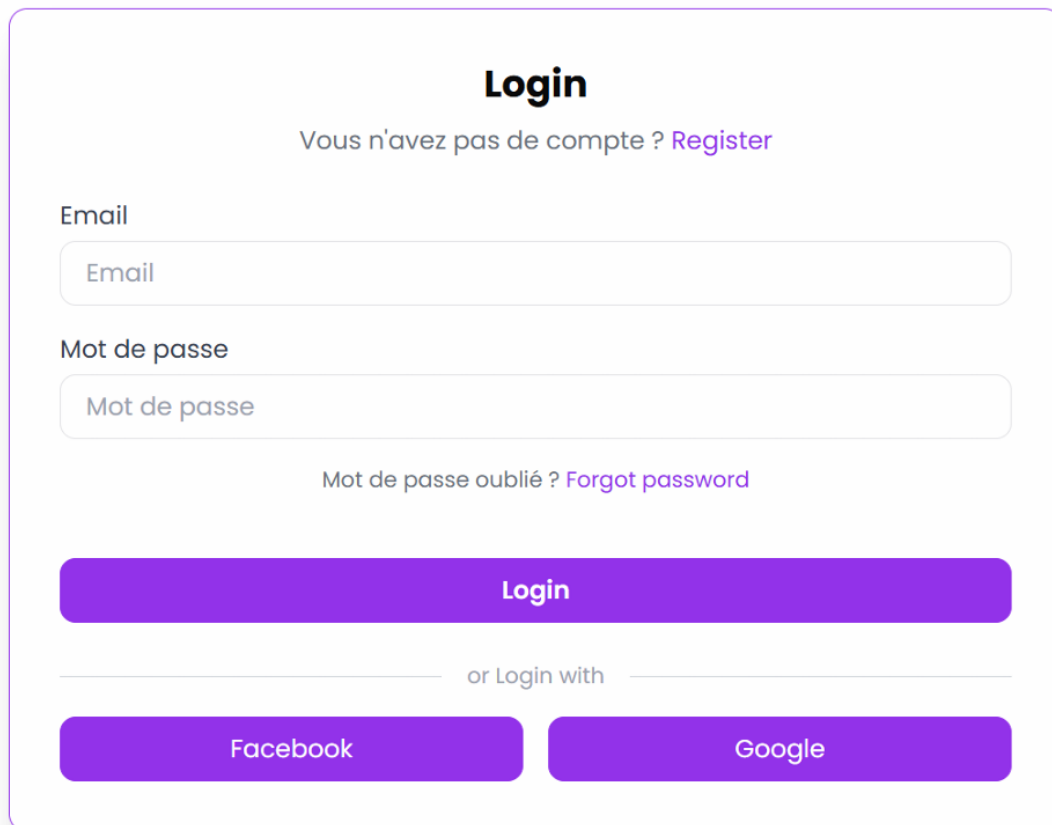


The image shows a registration form titled "Créer un compte" (Create an account) centered on a blue background with abstract wavy patterns. The form is a white rounded rectangle containing the following elements:

- Form title: "Créer un compte"
- Field 1: "Nom" (Name) with a red asterisk, a lock icon, and a text input field.
- Field 2: "Prénom" (First Name) with a red asterisk, a lock icon, and a text input field.
- Field 3: "Email" with a red asterisk, an email icon, and a text input field.
- Field 4: "Numéro de la carte d'identité" (ID Card Number) with a red asterisk, a lock icon, and a text input field.
- Field 5: "Numéro de téléphone" (Phone Number) with a red asterisk, a lock icon, and a text input field.
- Field 6: "Mot de passe" (Password) with a red asterisk, a lock icon, a text input field, and a visibility toggle icon.
- Button: "S'inscrire" (Sign up) in a blue rounded rectangle at the bottom.

**Figure 14: Interface S'inscrire pour le partenaire et l'administrateur**

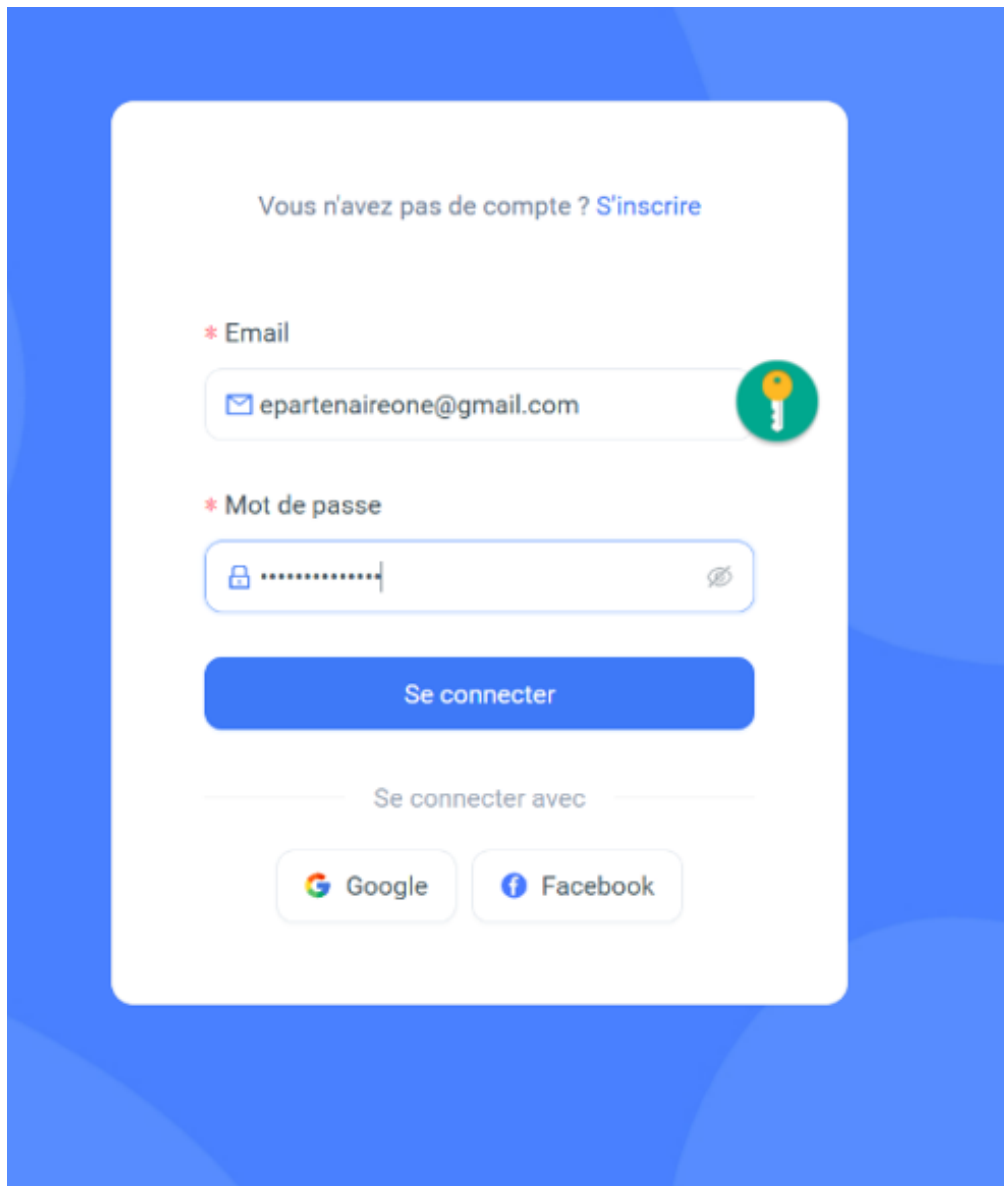
Après s'être inscrit, l'utilisateur peut se connecter à la plateforme des événements et fêtes à l'aide du formulaire de connexion suivant :



The image shows a login form with a purple border. At the top, the word "Login" is centered in a bold, black font. Below it, the text "Vous n'avez pas de compte ? Register" is displayed, with "Register" in purple. The form contains two input fields: "Email" and "Mot de passe" (Password), both with placeholder text. Below the password field, there is a link "Mot de passe oublié ? Forgot password" in purple. A large purple button labeled "Login" is positioned below the inputs. Underneath, the text "or Login with" is centered between two horizontal lines. At the bottom, there are two purple buttons labeled "Facebook" and "Google".

**Figure 15: Interface S'authentifier pour le client**

La figure suivante représente l'interface de connexion pour le partenaire et l'administrateur.



Vous n'avez pas de compte ? S'inscrire

\* Email

✉ epartenaireone@gmail.com

\* Mot de passe

🔒 .....| 🔍

Se connecter

Se connecter avec

🌐 Google

📘 Facebook

**Figure 16: Interface S'authentifier pour le partenaire et l'administrateur**

### 3.9 Sprint 2

Ce sprint dure quatre semaines, il comprend les tâches suivantes :

- Ajouter un client
- Modifier un client
- Supprimer un client
- Ajouter un partenaire
- Modifier un partenaire

-Supprimer un partenaire

-Ajouter un service

-Modifier un service

-Supprimer un service

Le résultat de ce sprint est un incrément livrable représentant la deuxième partie fonctionnelle de notre application. Dans ce qui suit, nous présentons le diagramme de cas d'utilisation correspondant, les descriptions textuelles des cas d'utilisation, les diagrammes de séquence, ainsi que le diagramme de classes participantes.

Enfin, nous illustrons les interfaces utilisateur développées au cours de ce sprint à l'aide de captures d'écran.

### 3.9.1 Descriptions textuelles

Les tableaux 6, 7 et 8, représentent les descriptions textuelles des fonctionnalités de gestion des clients dans le sprint 2.

**Tableau 6 : Description textuelle du cas d'utilisation Ajouter un client**

<b>Titre</b>	Ajouter un client
<b>Acteurs</b>	Administrateur
<b>Objectifs</b>	L'administrateur ajoute un nouveau client
<b>Pré conditions</b>	L'administrateur doit être authentifié
<b>Scénario nominal</b>	1/ L'administrateur s'authentifie 2/ L'administrateur accède à l'interface Clients 3/ L'administrateur clique sur le bouton Ajouter un client 4/L'administrateur remplit les champs du formulaire d'ajout 5/ L'administrateur clique sur le bouton Ajouter 6/Vérifier la syntaxe des données saisies 7/ Effectuer un traitement pour vérifier si le client existe déjà 8/ Envoyer les données pour les sauvegarder dans la BDD 9/ Nouveau client créé avec succès

<b>Scénario alternatif</b>	7/ Erreur dans la syntaxe des données, réessayer 8/ Erreur, le client existe déjà
<b>Post conditions</b>	Nouveau client ajouté avec succès

**Tableau 7 : Description textuelle du cas d'utilisation Modifier un client**

<b>Titre</b>	Modifier un client
<b>Acteurs</b>	Administrateur
<b>Objectifs</b>	Modifier les informations d'un client existant
<b>Pré conditions</b>	L'administrateur doit être authentifié Le client existe déjà
<b>Scénario nominal</b>	1/ L'administrateur s'authentifie 2/ L'administrateur accède à l'interface Clients 3/L'administrateur sélectionne un client et clique sur le bouton Modifier dans la case Actions 4/ Faire les modifications nécessaires 5/ Effectuer un traitement pour vérifier la syntaxe des nouvelles données 6/Envoyer les données pour les sauvegarder dans la BDD 7/ Effectuer un traitement pour vérifier si le client existe déjà 8/ Sauvegarder les nouvelles données dans la BDD 9/ Client modifié avec succès
<b>Scénario alternatif</b>	6/ Erreur dans la syntaxe des données 8/ Erreur, le client existe déjà
<b>Post conditions</b>	Client modifié avec succès

**Tableau 8 : Description textuelle du cas d'utilisation Supprimer un client**

<b>Titre</b>	Supprimer un client
<b>Acteurs</b>	Administrateur
<b>Objectifs</b>	Supprimer un client existant
<b>Pré conditions</b>	L'administrateur doit être authentifié Le client existe déjà
<b>Scénario nominal</b>	1/ L'administrateur s'authentifie 2/ L'administrateur accède à l'interface Clients 3/L'administrateur sélectionne un client et clique sur le bouton Supprimer dans la case Actions 4/ Confirmer la suppression 5/ Supprimer le client dans la BDD 6/ Client supprimé avec succès
<b>Scénario alternatif</b>	/
<b>Post conditions</b>	Client supprimé avec succès

Les tableaux 9, 10 et 11 représentent les descriptions textuelles des fonctionnalités de gestion des partenaires dans le sprint 2.

**Tableau 9 : Description textuelle du cas d'utilisation Ajouter un partenaire**

<b>Titre</b>	Ajouter un partenaire
<b>Acteurs</b>	Administrateur
<b>Objectifs</b>	L'administrateur ajoute un nouveau partenaire
<b>Pré conditions</b>	L'administrateur doit être authentifié
<b>Scénario nominal</b>	1/ L'administrateur s'authentifie 2/ L'administrateur accède à l'interface Partenaires 3/ L'administrateur clique sur le bouton Ajouter un partenaire 4/Le partenaire remplit les champs du formulaire d'ajout

	<p>5/ Effectuer un traitement pour vérifier la syntaxe des données saisies</p> <p>6/ Envoyer les données pour les sauvegarder dans la BDD</p> <p>7/ Effectuer un traitement pour vérifier si le partenaire existe déjà</p> <p>8/ Sauvegarder les nouvelles données dans la BDD</p> <p>8/ Partenaire modifié avec succès</p>
<b>Scénario alternatif</b>	<p>6/ Erreur dans la syntaxe des données</p> <p>8/ Erreur, le partenaire existe déjà</p>
<b>Post conditions</b>	Partenaire ajouté avec succès

**Tableau 10 : Description textuelle du cas d'utilisation Modifier un partenaire**

<b>Titre</b>	Modifier un partenaire
<b>Acteurs</b>	Administrateur
<b>Objectifs</b>	Modifier les informations d'un partenaire existant
<b>Pré conditions</b>	<p>L'administrateur doit être authentifié</p> <p>Le partenaire existe déjà</p>
<b>Scénario nominal</b>	<p>1/ L'administrateur s'authentifie</p> <p>2/ L'administrateur accède à l'interface Partenaires</p> <p>3/ L'administrateur sélectionne un partenaire et clique sur le bouton Modifier dans la case Actions.</p> <p>4/ Faire les modifications nécessaires</p> <p>5/ Effectuer un traitement pour vérifier la syntaxe des nouvelles données</p> <p>6/ Effectuer un traitement pour vérifier si le partenaire existe déjà</p> <p>7/ Envoyer les données pour les sauvegarder dans la BDD</p> <p>8/ Partenaire modifié avec succès</p>

<b>Scénario alternatif</b>	6/ Erreur dans la syntaxe des données, réessayer 7/ Erreur, le partenaire existe déjà
<b>Post conditions</b>	Client modifié avec succès

**Tableau 11 : Description textuelle du cas d'utilisation Supprimer un partenaire**

<b>Titre</b>	Supprimer un partenaire
<b>Acteurs</b>	Administrateur
<b>Objectifs</b>	Supprimer un partenaire existant
<b>Pré conditions</b>	L'administrateur doit être authentifié Le partenaire existe déjà
<b>Scénario nominal</b>	1/ L'administrateur s'authentifie 2/ L'administrateur accède à l'interface Partenaires 3/L'administrateur sélectionne un partenaire et clique sur le bouton Supprimer dans la case Actions 4/ Confirmer la suppression 5/ Supprimer le partenaire dans la BDD 6/ Partenaire supprimé avec succès
<b>Scénario alternatif</b>	/
<b>Post conditions</b>	Partenaire supprimé avec succès

Les tableaux 12, 13 et 14, représentent les descriptions textuelles des fonctionnalités de gestion des services dans le sprint 2.

**Tableau 12 : Description textuelle du cas d'utilisation Ajouter un service**

<b>Titre</b>	Ajouter un service
<b>Acteurs</b>	Partenaire
<b>Objectifs</b>	Le partenaire propose un nouveau service à des clients
<b>Pré conditions</b>	Le partenaire doit être authentifié
<b>Scénario nominal</b>	1/ Le partenaire s'authentifie 2/ Le partenaire accède à l'interface Mes Services

	<p>3/ Le partenaire clique sur le bouton Ajouter un service</p> <p>4/Le partenaire remplit les champs du formulaire d'ajout du service</p> <p>5/ Effectuer un traitement pour vérifier la syntaxe des données</p> <p>6/ Enregistrer les données dans la BDD</p> <p>7/ Nouveau service créé avec succès</p>
<b>Scénario alternatif</b>	6/a/ Erreur dans la syntaxe des données, réessayer
<b>Post conditions</b>	Service ajouté avec succès

**Tableau 13 : Description textuelle du cas d'utilisation Modifier un service**

<b>Titre</b>	Modifier un service
<b>Acteurs</b>	Partenaire
<b>Objectifs</b>	Modifier les informations d'un service existant
<b>Pré conditions</b>	<p>Le partenaire doit être authentifié</p> <p>Le service existe déjà</p>
<b>Scénario nominal</b>	<p>1/ Le partenaire s'authentifie</p> <p>2/ Le partenaire accède à l'interface Mes services</p> <p>3/Le partenaire sélectionne un service et clique sur le bouton Modifier dans la case Actions</p> <p>4/ Faire les modifications nécessaires</p> <p>5/ Effectuer un traitement pour vérifier la syntaxe des données</p> <p>6/ Sauvegarder les nouvelles données dans la BDD</p> <p>7/ Service modifié avec succès</p>
<b>Scénario alternatif</b>	6/ Erreur dans la syntaxe des données, réessayer
<b>Post conditions</b>	Service modifié avec succès

**Tableau 14 : Description textuelle du cas d'utilisation Supprimer un service**

<b>Titre</b>	Supprimer un service
<b>Acteurs</b>	Partenaire
<b>Objectifs</b>	Supprimer un service existant
<b>Pré conditions</b>	Le partenaire doit être authentifié Le service existe déjà
<b>Scénario nominal</b>	1/ Le partenaire s'authentifie 2/ Le partenaire accède à l'interface Mes services 3/Le partenaire sélectionne un service et clique sur le bouton Supprimer dans la case Actions 4/ Confirmer la suppression 5/ Supprimer le service dans la BDD 7/ Service supprimé avec succès
<b>Scénario alternatif</b>	/
<b>Post conditions</b>	Service supprimé avec succès

### 3.9.2 Diagramme de cas d'utilisation du sprint 2

Dans la figure 17, on regroupe les informations des tableaux précédents qui décrivent les cas d'utilisation du sprint 2.

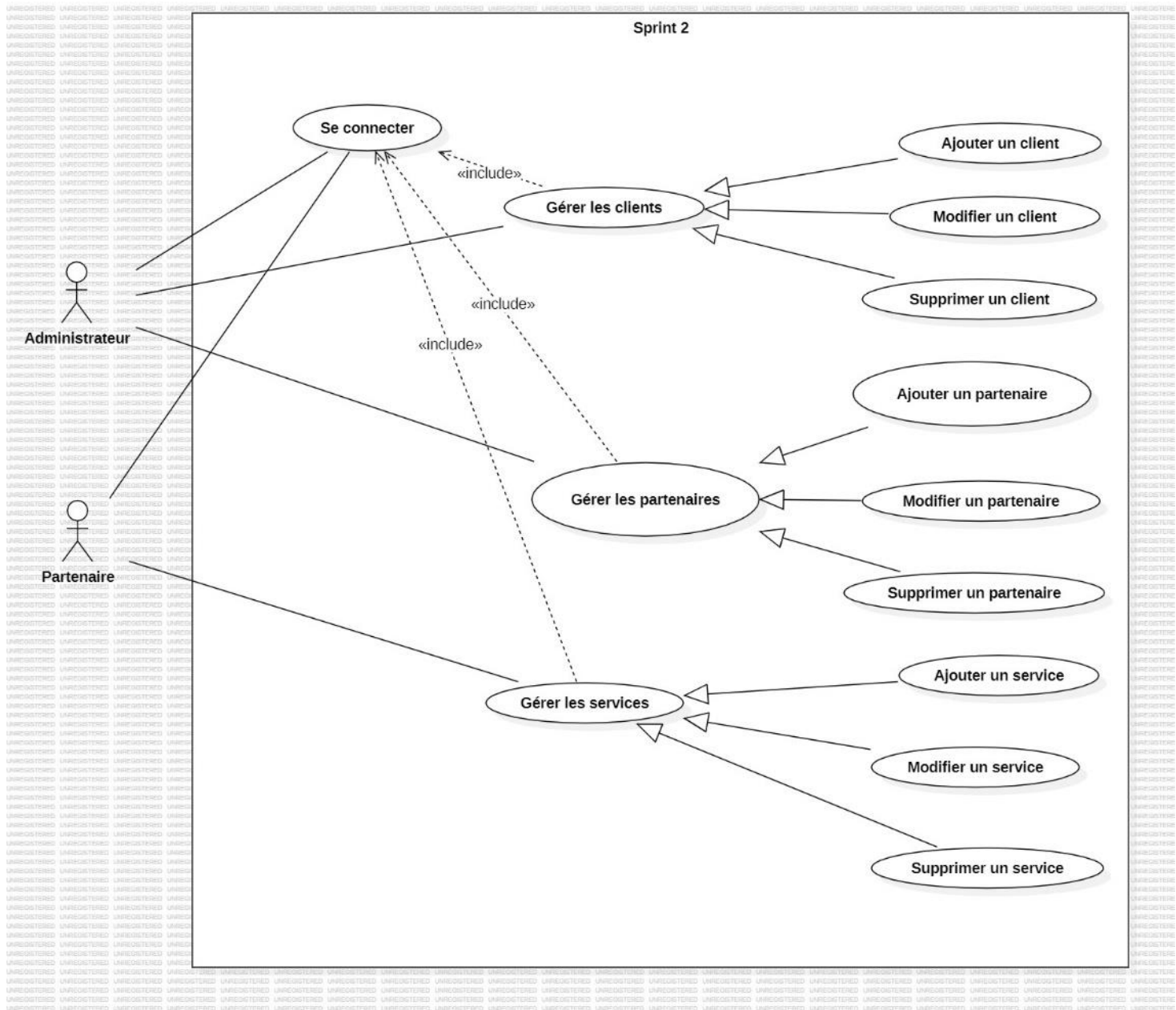
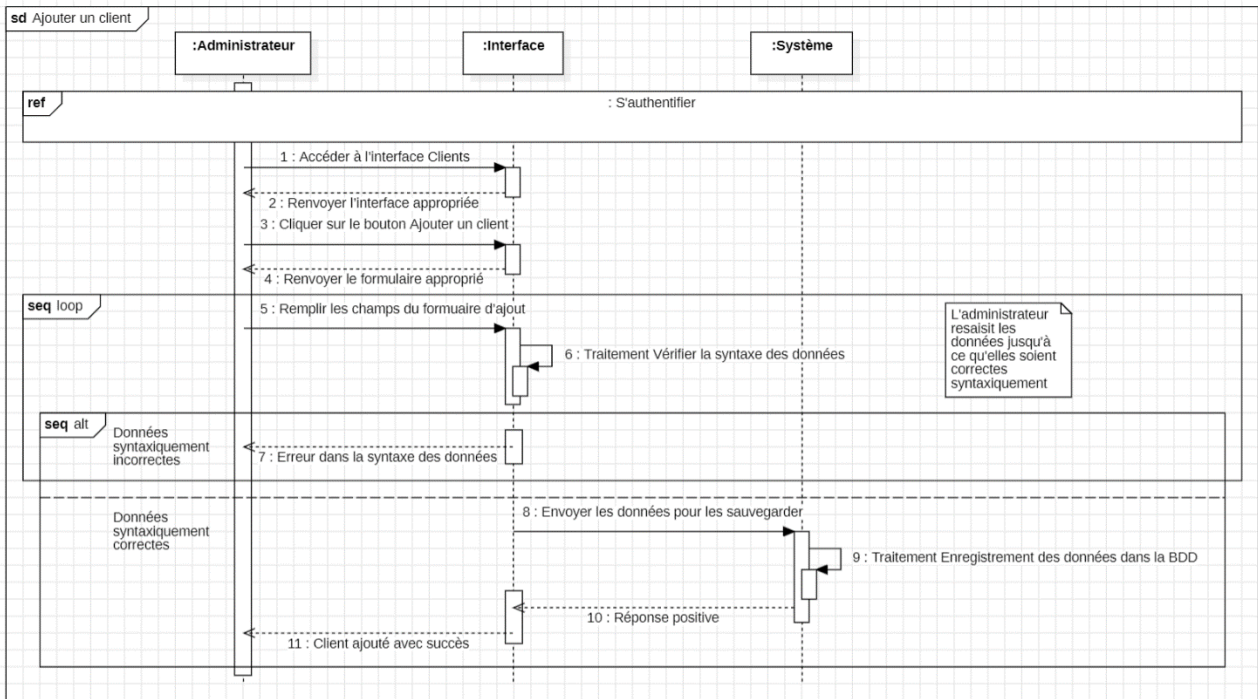


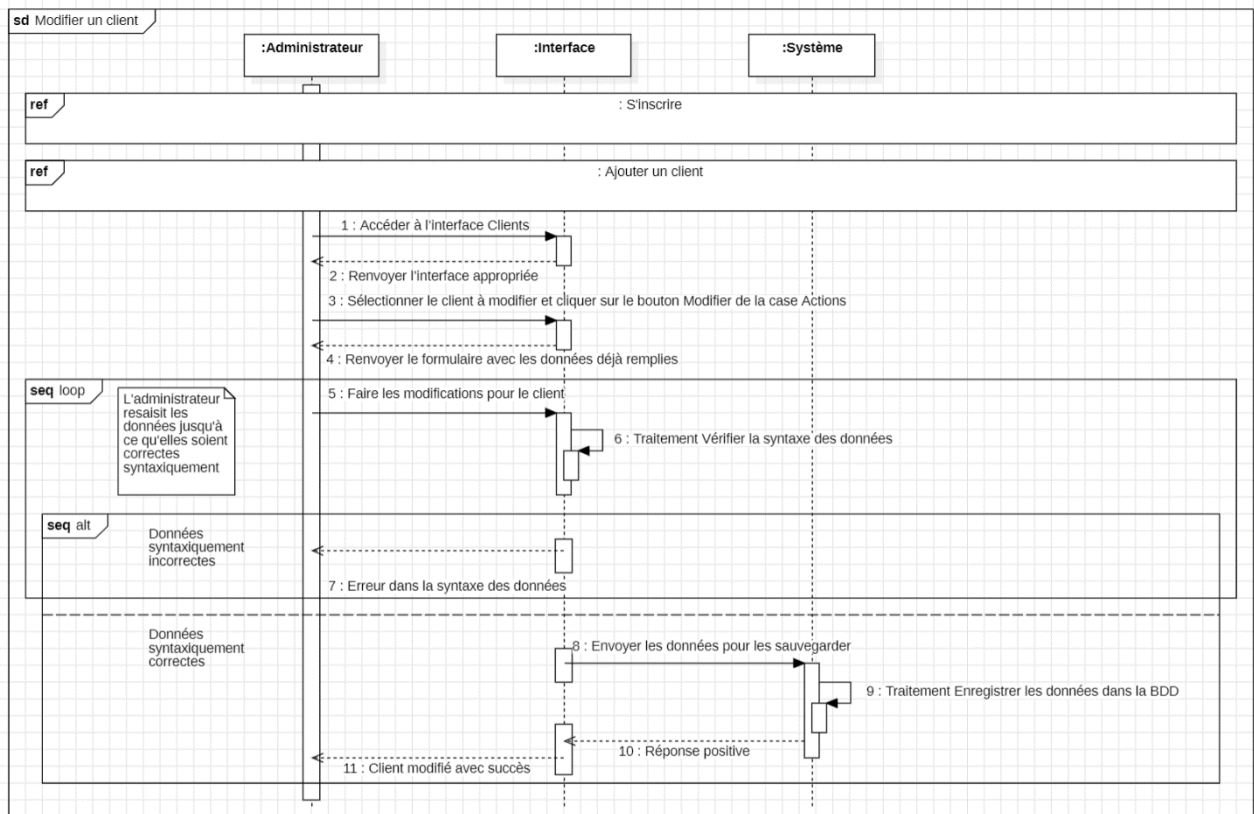
Figure 17: Diagramme de cas d'utilisation du sprint 2

### 3.9.3 Diagrammes de séquence

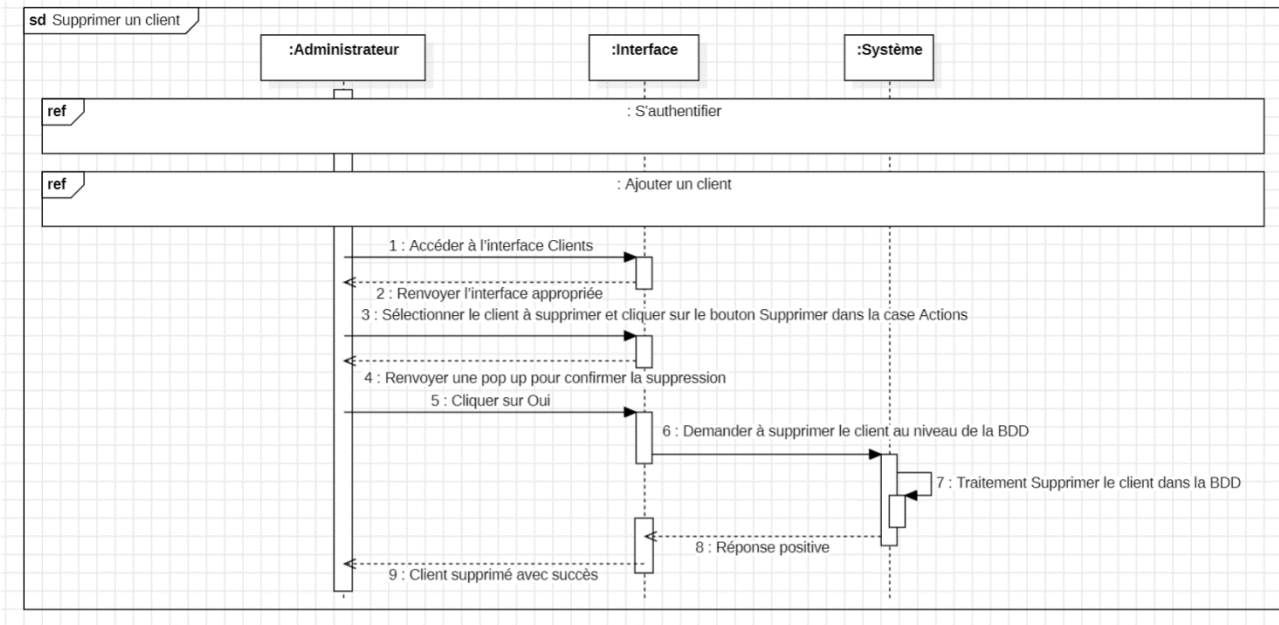
Les figures 18, 19 et 20 représentent respectivement les diagrammes de séquence de (ajouter un client, modifier un client et supprimer un client).



**Figure 18: Diagramme de séquence du cas d'utilisation Ajouter un client**

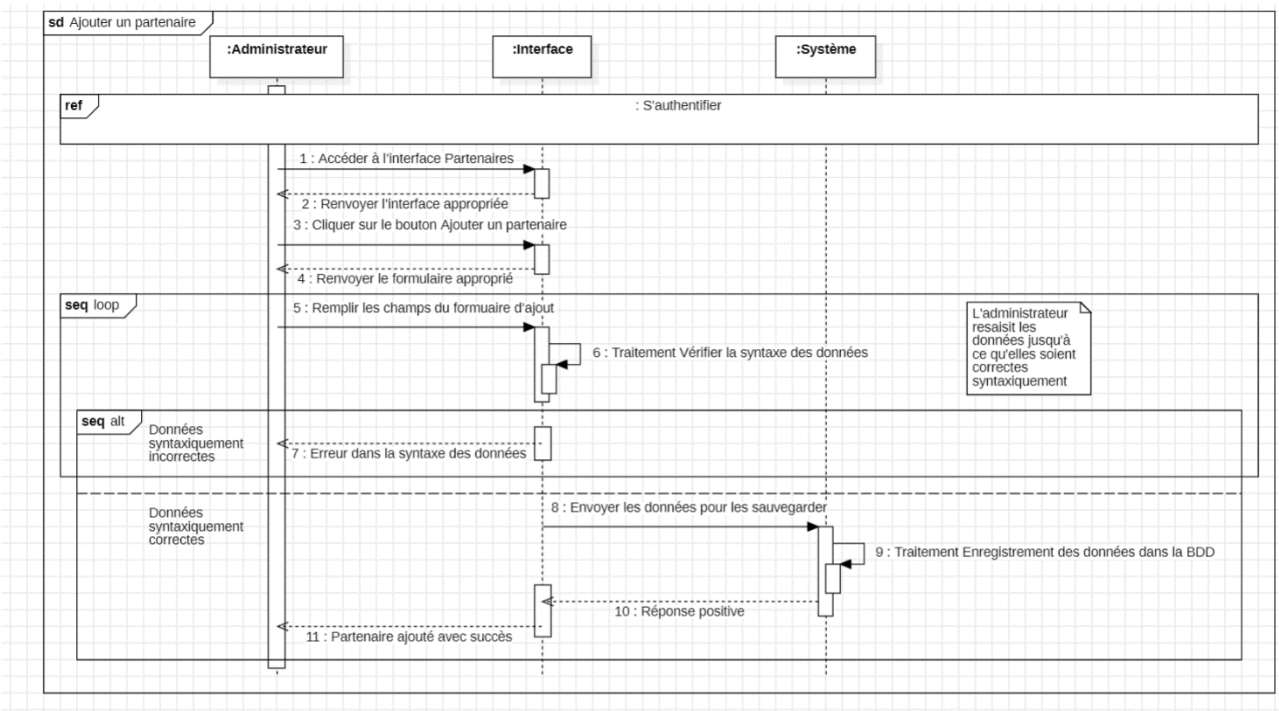


**Figure 19 Diagramme de séquence du cas d'utilisation Modifier un client**

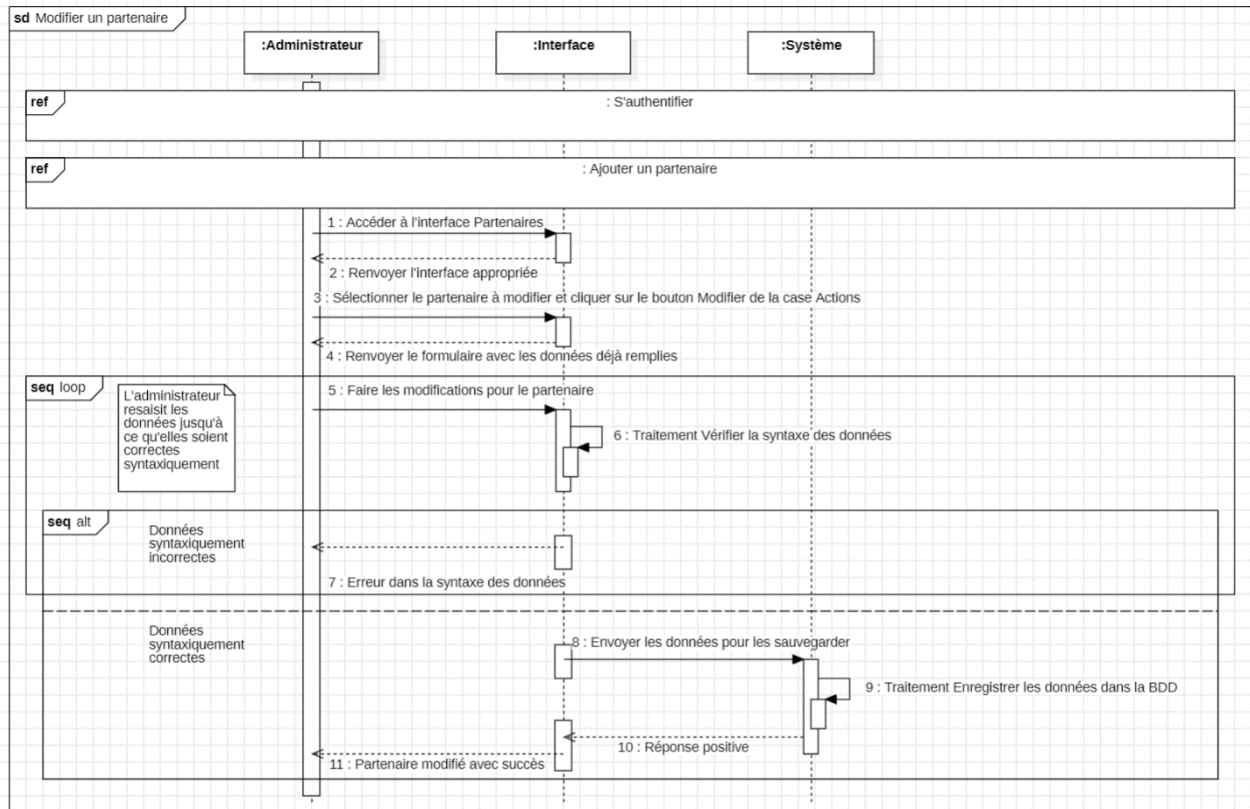


**Figure 20 Diagramme de séquence du cas d'utilisation Supprimer un client**

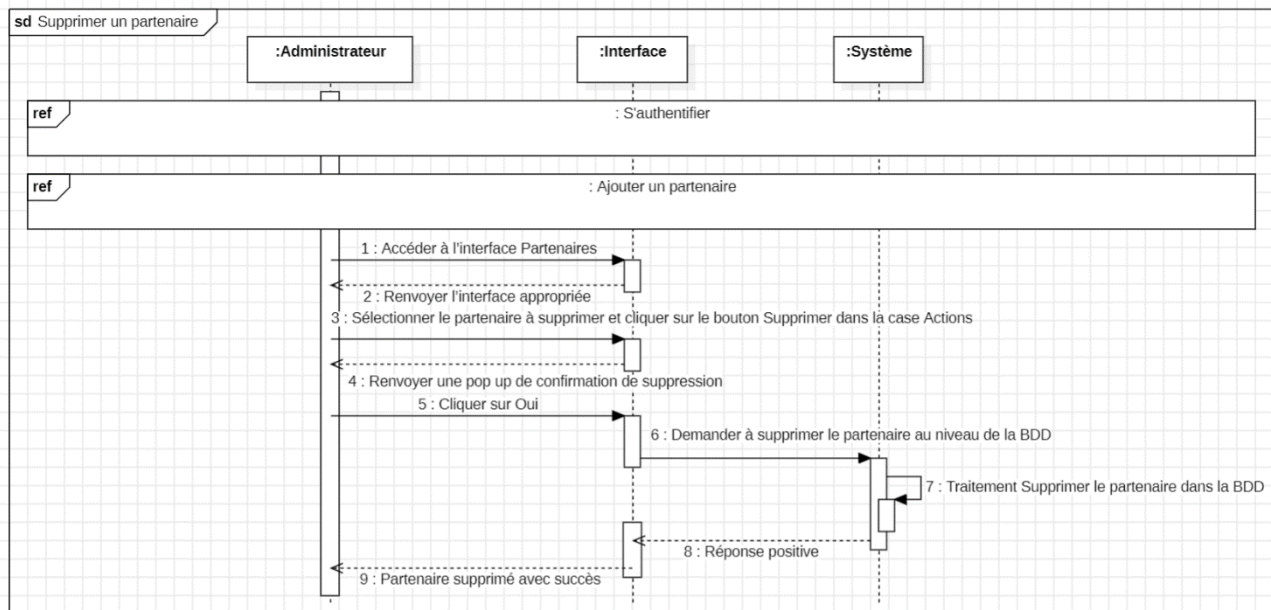
Les figures 21, 22 et 23 représentent respectivement les diagrammes de séquence de (ajouter un partenaire, modifier un partenaire et supprimer un partenaire).



**Figure 21: Diagramme de séquence du cas d'utilisation Ajouter un partenaire**

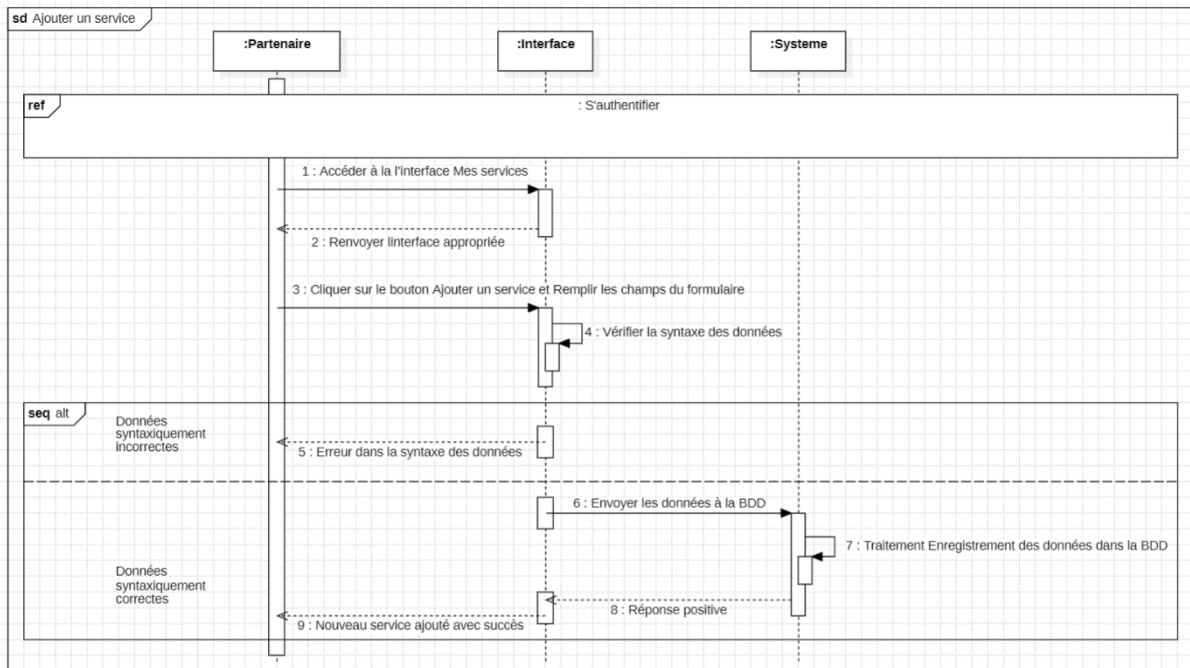


**Figure 22: Diagramme de séquence du cas d'utilisation Modifier un partenaire**

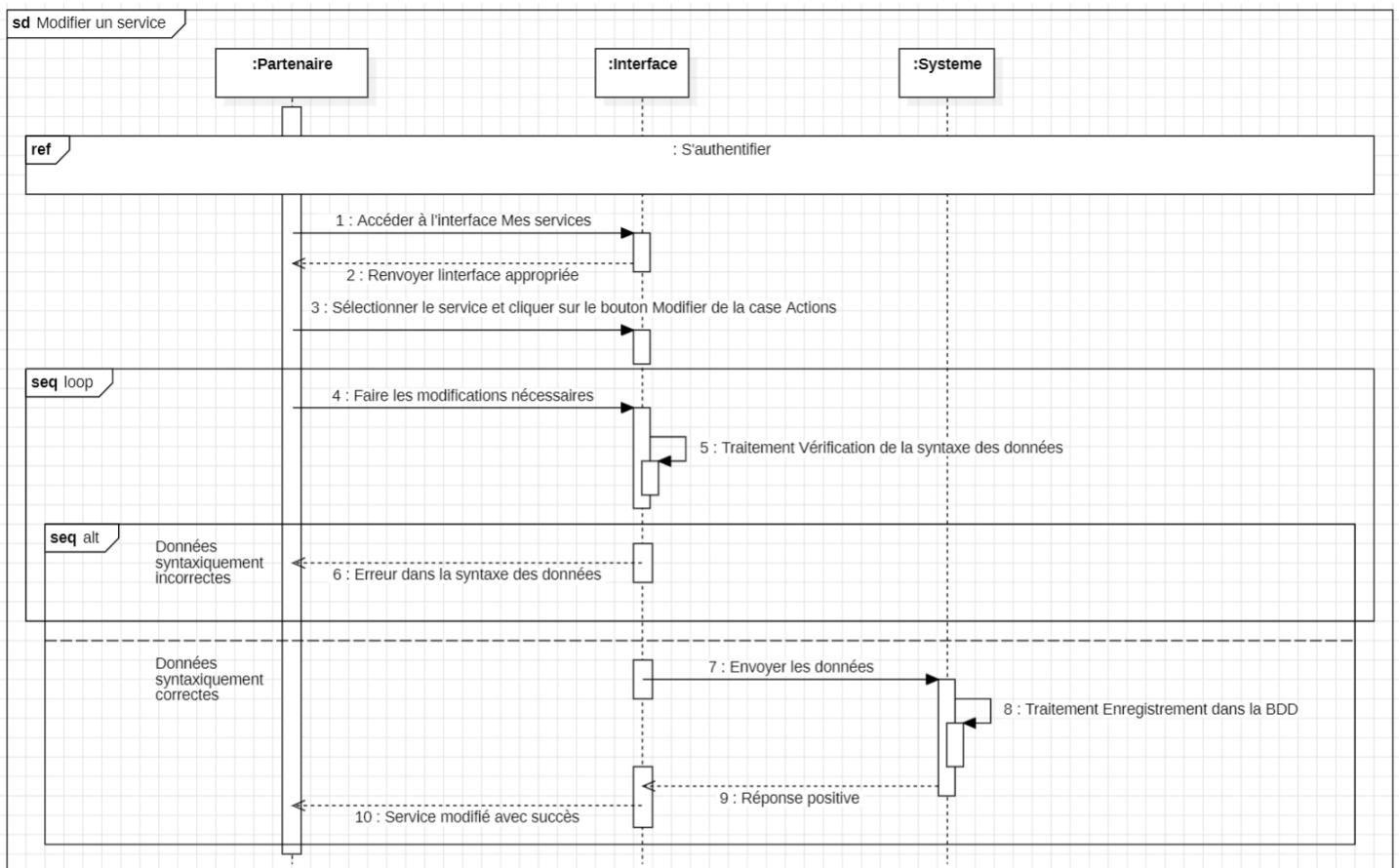


**Figure 23: Diagramme de séquence du cas d'utilisation Supprimer un partenaire**

Les figures 24, 25 et 26 représentent respectivement les diagrammes de séquence de (ajouter un service, modifier un service et supprimer un service).



**Figure 24: Diagramme de séquence du cas d'utilisation Ajouter un service**



**Figure 25: Diagramme de séquence du cas d'utilisation Modifier un service**

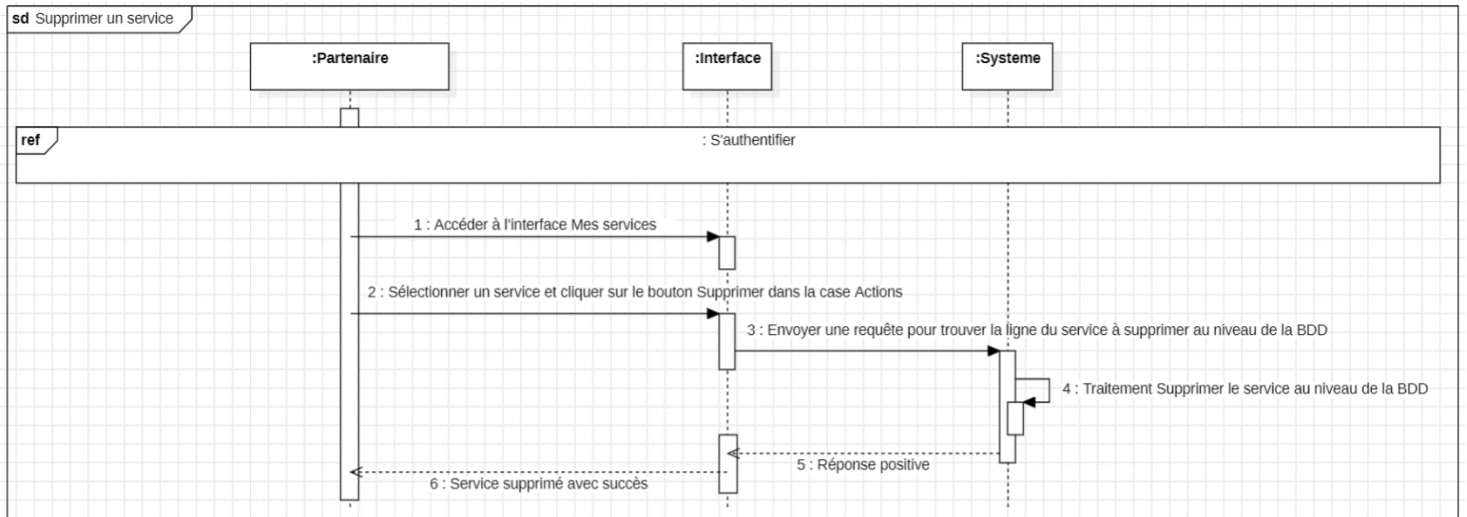


Figure 26: Diagramme de séquence du cas d'utilisation Supprimer un service

### 3.9.4 Diagramme de classes participantes

La figure 27 représente un diagramme où on peut voir les classes participantes dans le sprint 2 et comment elles sont liées.

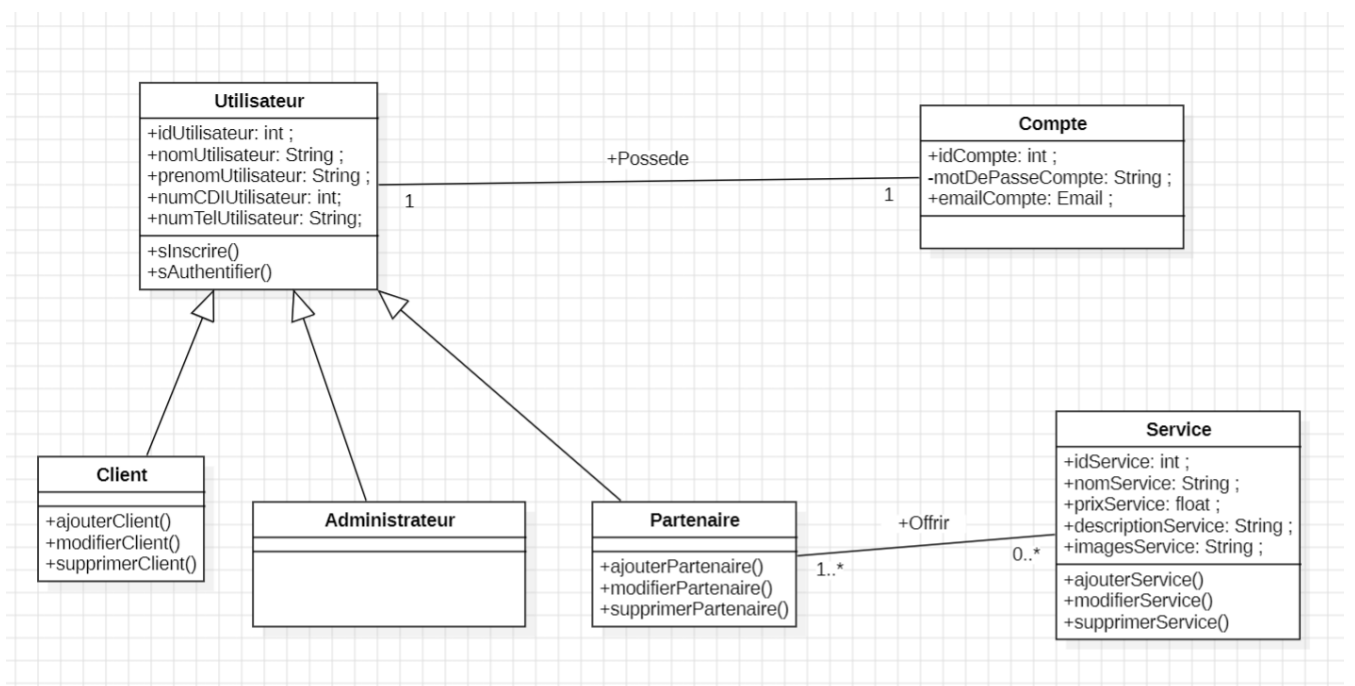


Figure 27: Diagramme de classes participantes du sprint 2

### 3.9.5 Interfaces utilisateur

Les figures 28, 29 et 30 représentent les interfaces d'ajout, modification et suppression d'un partenaire.

## Liste des partenaires

[Ajouter un partenaire](#)

### Ajouter un partenaire

nom
prenom
email
password
numCDI
numTel
<input type="button" value="Enregistrer"/> <input type="button" value="Annuler"/>

**Figure 28: Interface Ajouter un partenaire**

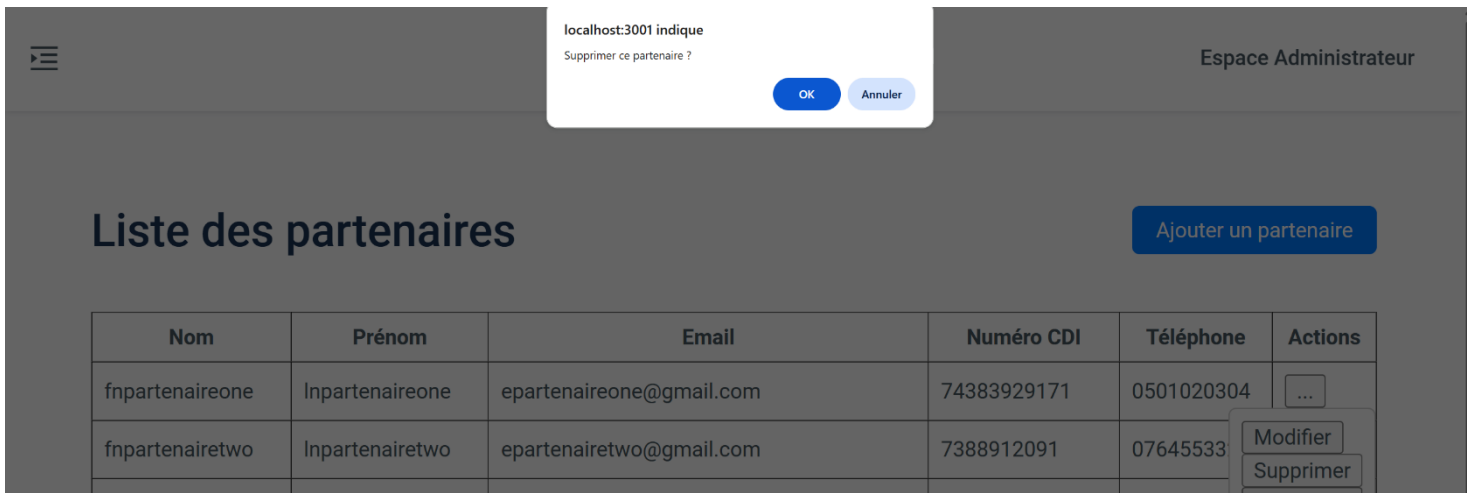
## Liste des partenaires

[Ajouter un partenaire](#)

### Modifier le partenaire

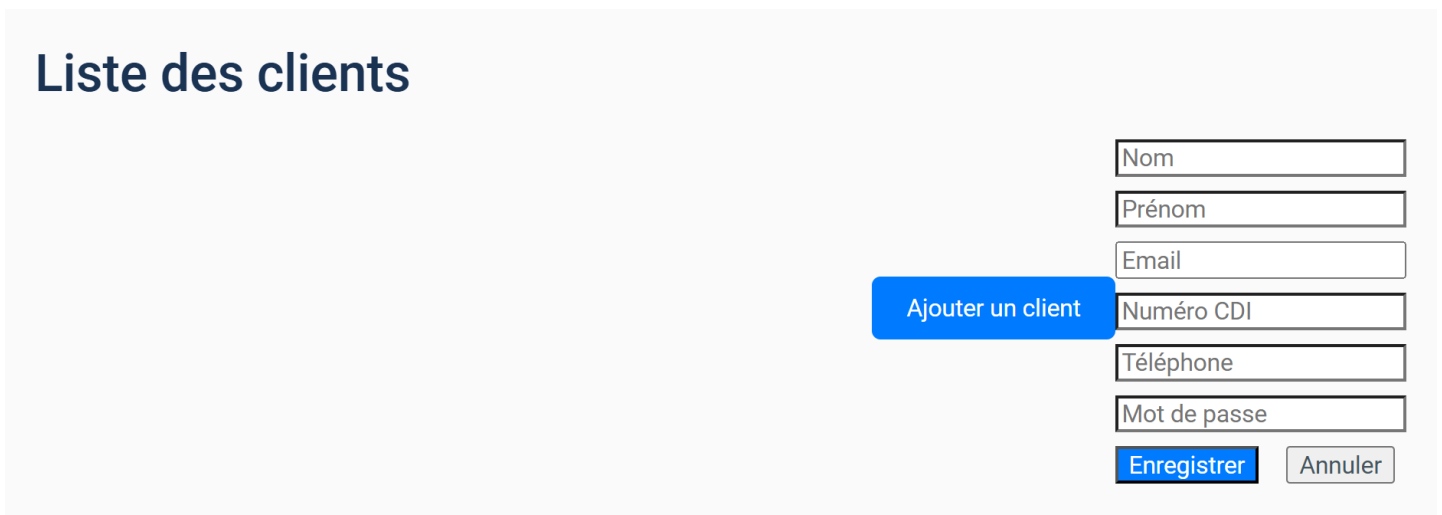
fnpartenaireone
lnpartenaireone
epartenaireone@gmail.com
74383929171
0501020304
<input type="button" value="Modifier"/> <input type="button" value="Annuler"/>

**Figure 29: Interface Modifier un partenaire**

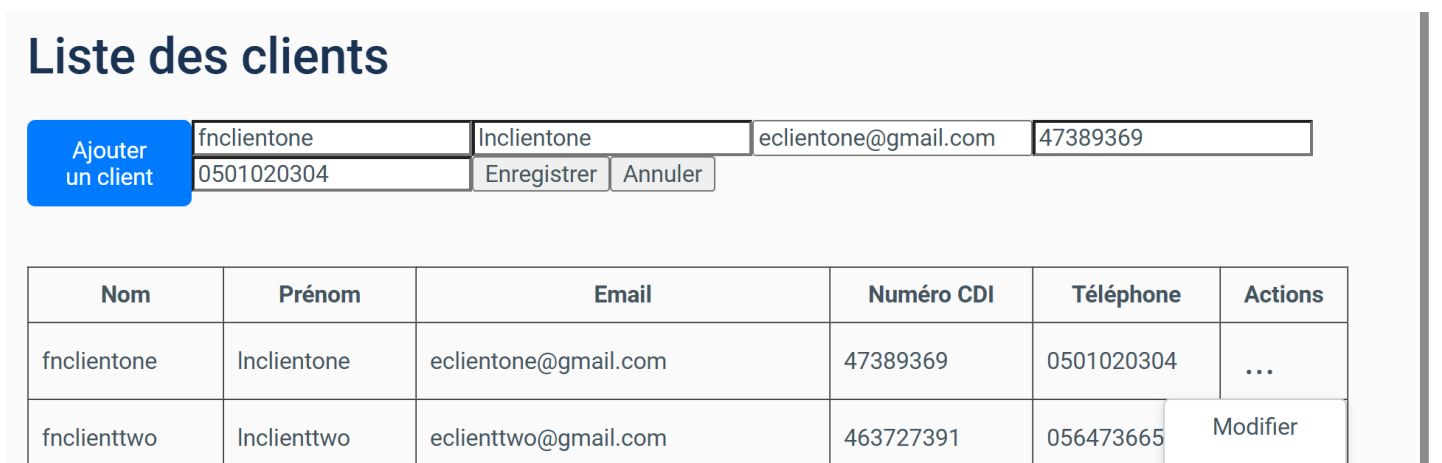


**Figure 30: Interface Supprimer un partenaire**

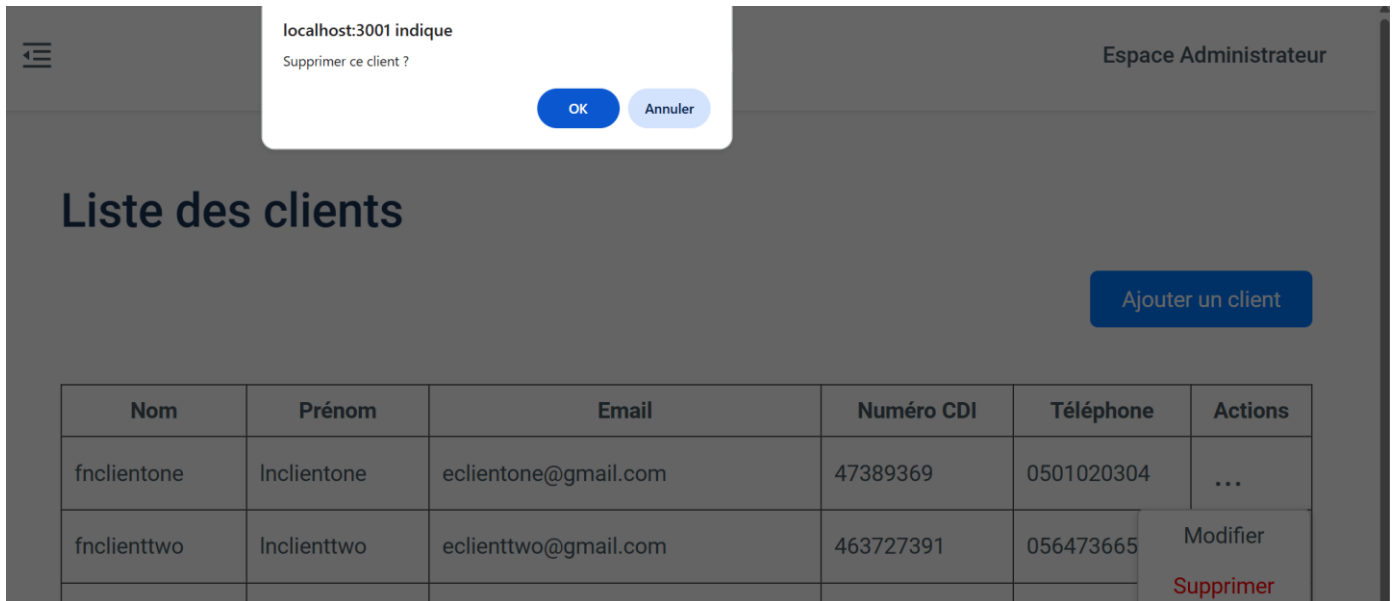
Les figures 31, 32 et 33 représentent les interfaces d'ajout, de modification et de suppression d'un client.



**Figure 31: Interface Ajouter un client**

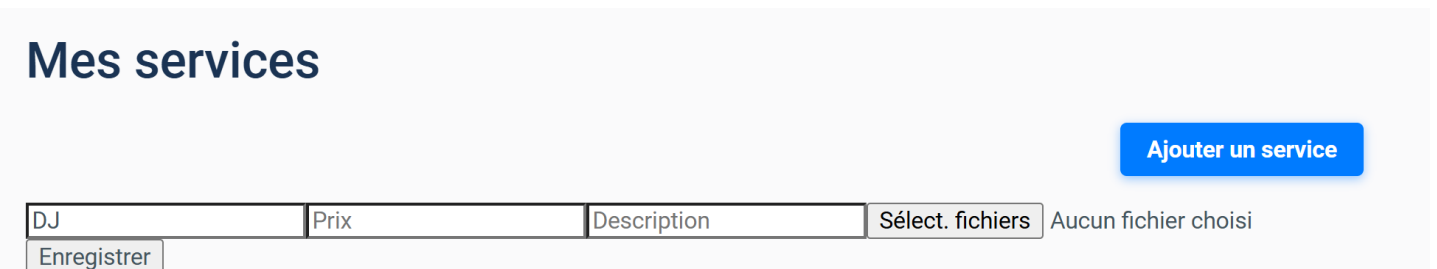


**Figure 32: Interface Modifier un client**



**Figure 33: Interface Supprimer un client**

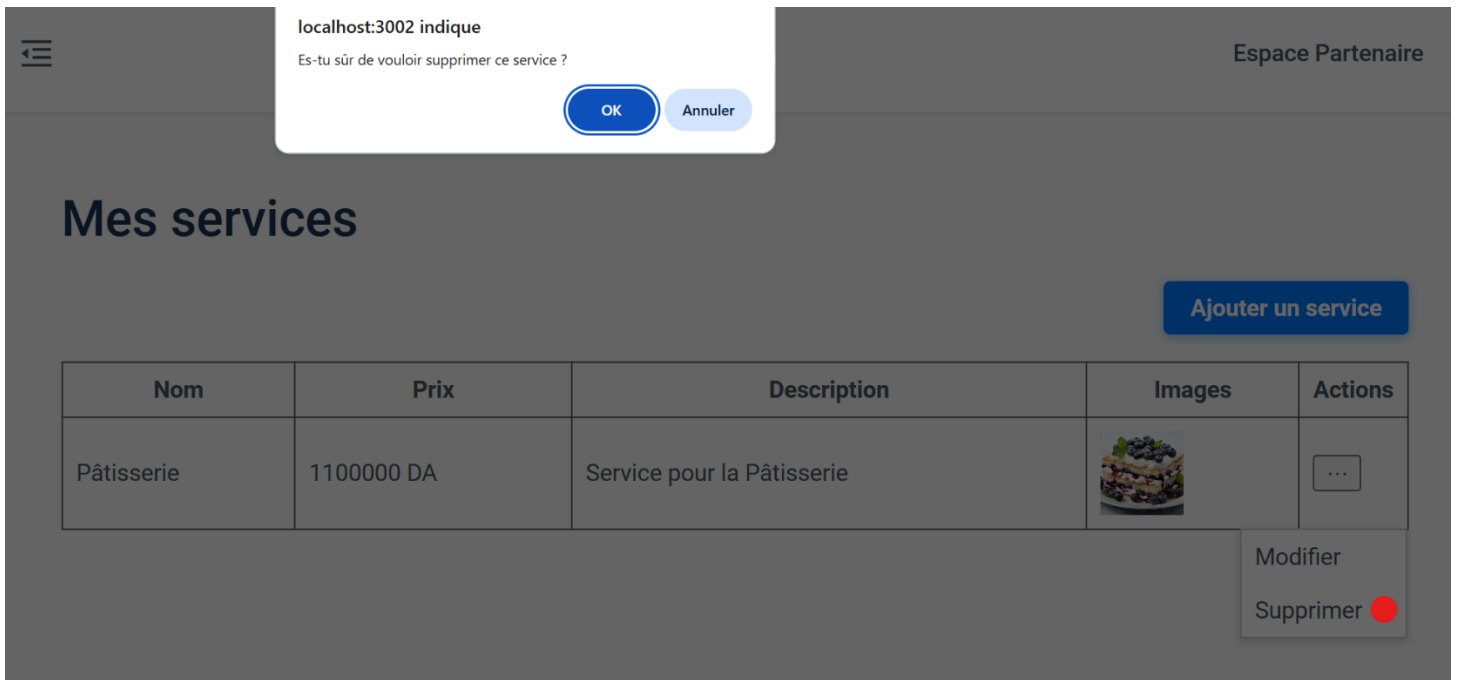
Les figures 34, 35 et 36 représentent les interfaces d'ajout, modification et suppression d'un service.



**Figure 34: Interface Ajouter un service**



**Figure 35: Interface Modifier un service**



**Figure 36: Interface Supprimer un service**

### 3.10 Sprint 3

Ce sprint dure deux semaines, il comprend les tâches suivantes :

- Ajouter une réservation
- Modifier une réservation
- Supprimer une réservation

Ce sprint a permis de produire un livrable représentant la troisième fonctionnalité clé de notre application.

Dans les sections qui suivent, nous exposons le diagramme des cas d'utilisation associé, les descriptions textuelles des différents scénarios d'utilisation, les diagrammes de séquence, ainsi que le diagramme des classes participantes.

Pour terminer, les interfaces utilisateurs développées au cours de ce sprint sont présentées à travers des illustrations issues de l'application.

#### 3.10.1 Descriptions textuelles

Les tableaux 15, 16 et 17 représentent respectivement les descriptions textuelles des cas d'utilisation de (ajouter, modifier et supprimer une réservation).

**Tableau 15 : Description textuelle du cas d'utilisation Ajouter une réservation**

<b>Titre</b>	Ajouter une réservation
<b>Acteurs</b>	Client
<b>Objectifs</b>	Ajouter une nouvelle réservation pour un prochain événement
<b>Pré conditions</b>	Le client doit être authentifié Les partenaires ont déjà proposé des services
<b>Scénario nominal</b>	1/ Le client s'authentifie 2/ Le client accède à l'interface Mes réservations 3/ Le client clique sur le bouton Ajouter une réservation 4/ Le client remplit les champs du formulaire d'ajout de la réservation 5/ Effectuer un traitement pour vérifier la syntaxe des données 6/ Effectuer un traitement pour vérifier si la réservation existe déjà 7/ Enregistrer les données dans la BDD 8/ Réservation ajoutée avec succès
<b>Scénario alternatif</b>	6/ Erreur dans la syntaxe des données, réessayer 7/ Erreur, la réservation existe déjà
<b>Post conditions</b>	Réservation ajoutée avec succès

**Tableau 16 : Description textuelle du cas d'utilisation Modifier une réservation**

<b>Titre</b>	Modifier une réservation
<b>Acteurs</b>	Client
<b>Objectifs</b>	Modifier les informations d'une réservation existante
<b>Pré conditions</b>	Le client doit être authentifié La réservation existe déjà
<b>Scénario nominal</b>	1/ Le client s'authentifie 2/ Le client accède à l'interface Réservations 3/ Le client sélectionne une réservation et clique sur le bouton Modifier de la case Actions 4/ Faire les modifications nécessaires 5/ Effectuer un traitement pour vérifier la syntaxe des données 6/ Effectuer un traitement pour vérifier si la réservation existe déjà 7/ Enregistrer les données dans la BDD 8/ Réservation modifiée avec succès
<b>Scénario alternatif</b>	6/ Erreur dans la syntaxe des données, réessayer 7/ Erreur, la réservation existe déjà
<b>Post conditions</b>	Réservation modifiée avec succès

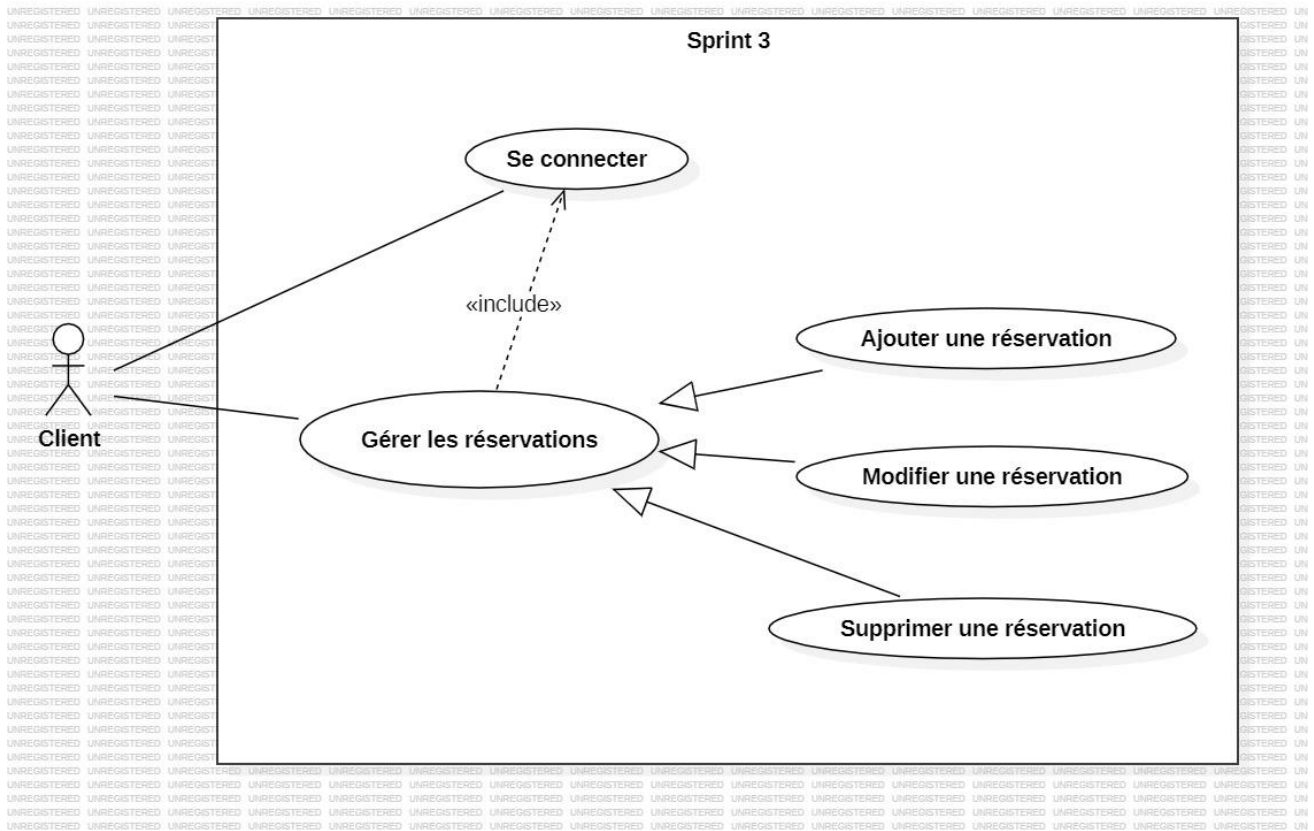
**Tableau 17 : Description textuelle du cas d'utilisation Supprimer une réservation**

<b>Titre</b>	Supprimer une réservation
<b>Acteurs</b>	Client
<b>Objectifs</b>	Supprimer une réservation déjà existante
<b>Pré conditions</b>	Le client doit être authentifié La réservation existe déjà
<b>Scénario nominal</b>	1/ Le client s'authentifie 2/ Le client accède à l'interface Réservations

	<p>3/ Le client sélectionne une réservation et clique sur le bouton Supprimer de la case Actions</p> <p>4/ Confirmer la suppression</p> <p>5/ Supprimer la réservation dans la BDD</p> <p>7/ Réservation supprimée avec succès</p>
<b>Scénario alternatif</b>	/
<b>Post conditions</b>	Réservation supprimée avec succès

### 3.10.2 Diagramme de cas d'utilisation du sprint 3

Dans la figure 37, on regroupe les informations des tableaux précédents qui décrivent les cas d'utilisation du sprint 3.



**Figure 37 : Diagramme de cas d'utilisation du sprint 3**

### 3.10.3 Diagrammes de séquence

Les figures 38, 39 et 40 représentent respectivement les diagrammes de séquence des cas d'utilisation ajouter, modifier et supprimer une réservation.

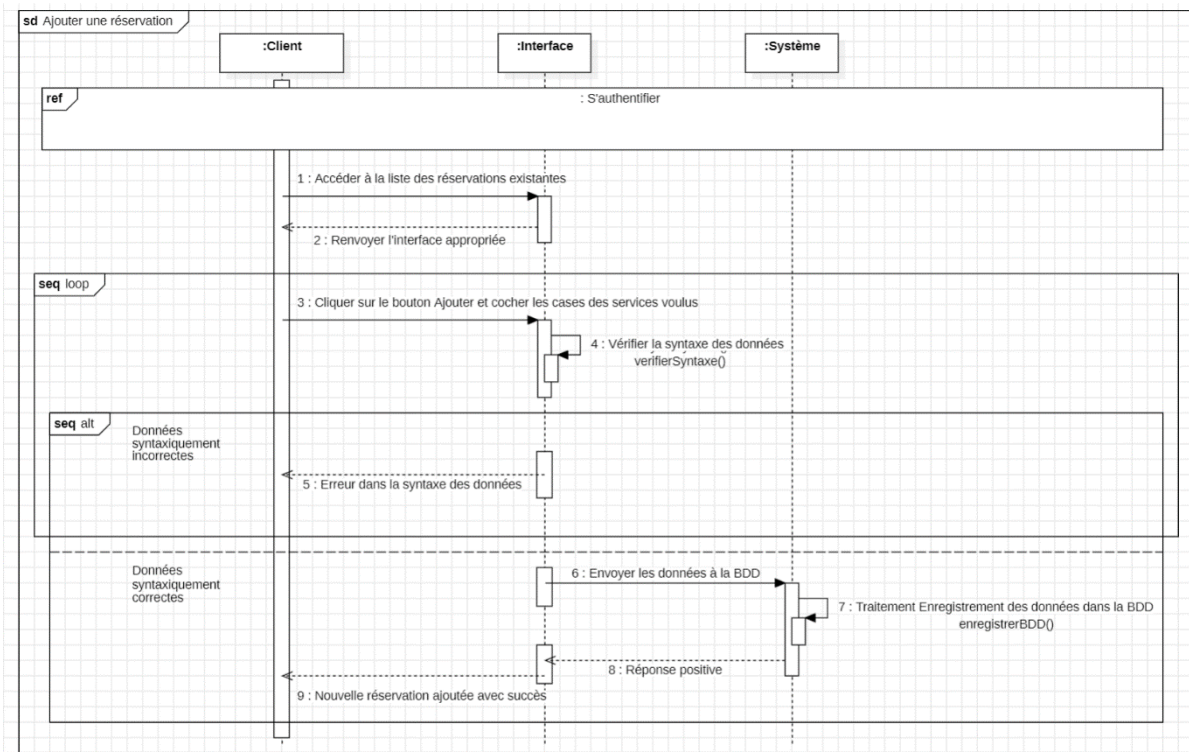


Figure 38: Diagramme de séquence du cas d'utilisation Ajouter une réservation

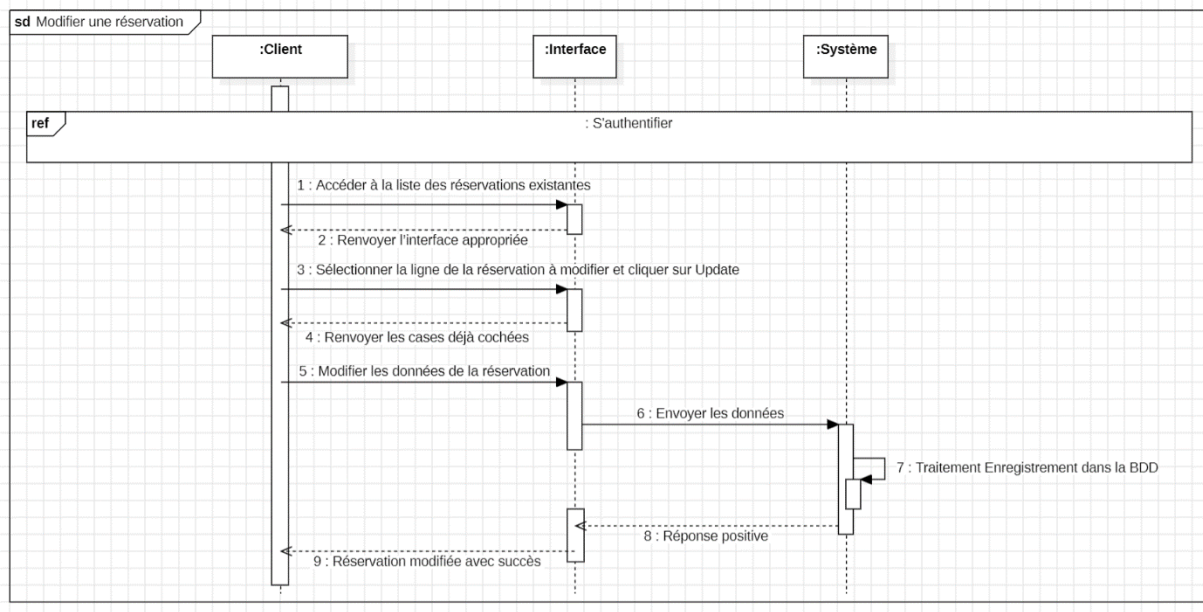
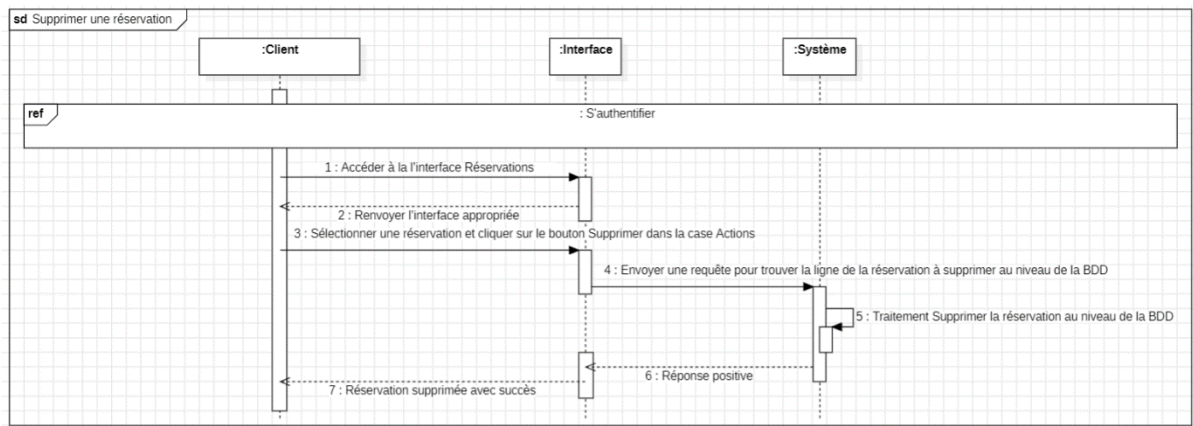


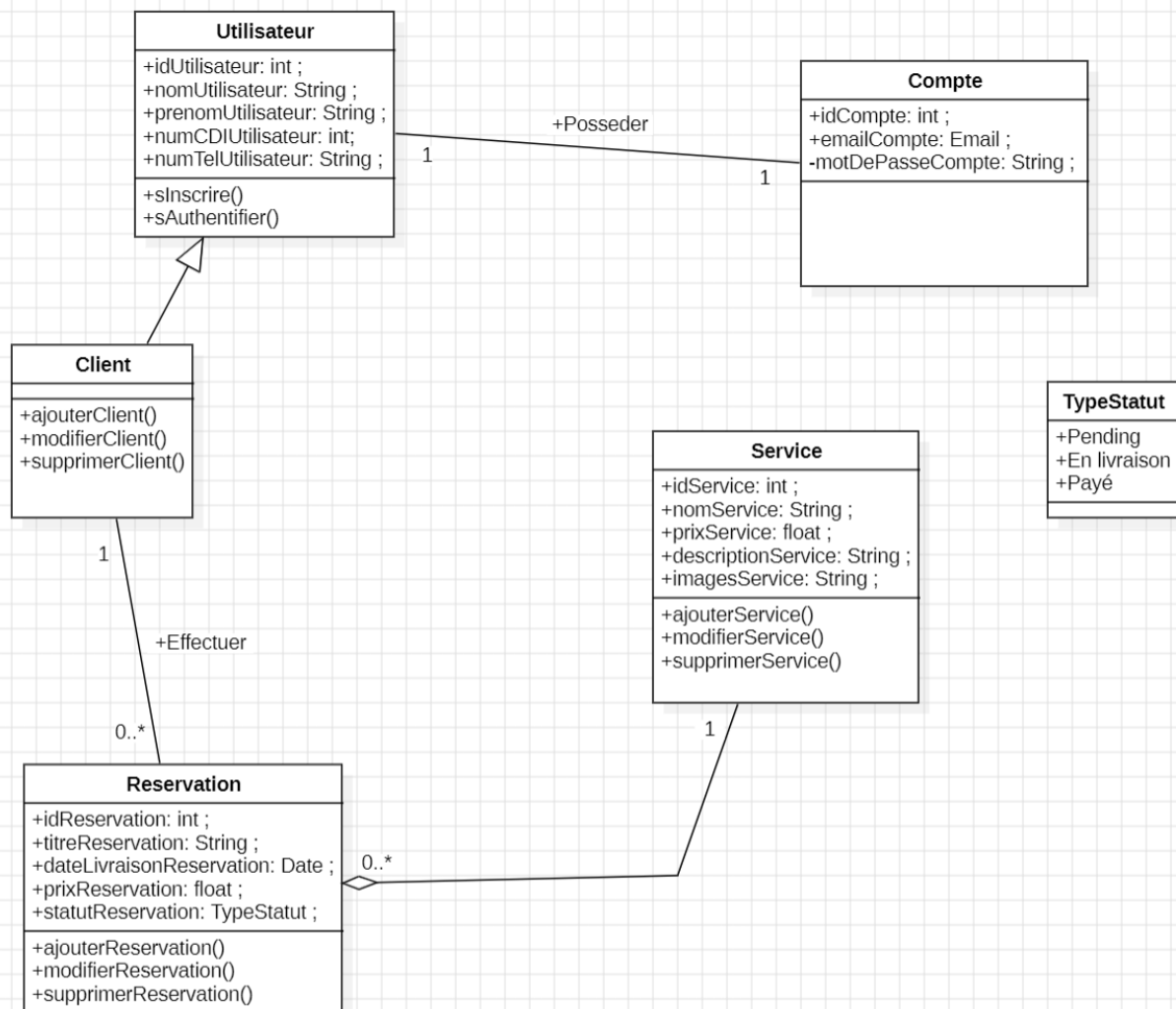
Figure 39: Diagramme de séquence du cas d'utilisation Modifier une réservation



**Figure 40: Diagramme de séquence du cas d'utilisation Supprimer une réservation**

### 3.10.4 Diagramme de classes participantes

La figure 41 représente les classes participantes du sprint 3.



**Figure 41: Diagramme de classes participantes du sprint 3**

### 3.10.5 Interfaces utilisateur

Les figures 42, 43 et 44 représentent les interfaces d'ajout, modification et suppression d'une réservation.

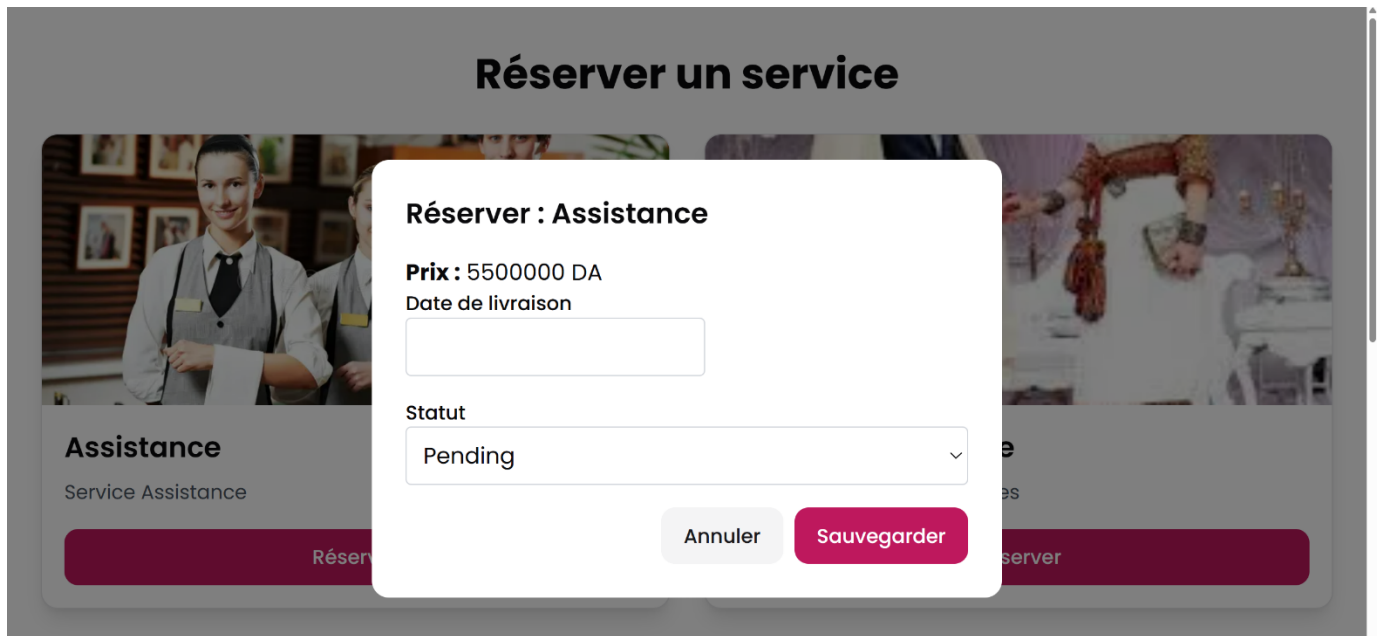


Figure 42: Interface Ajouter une réservation

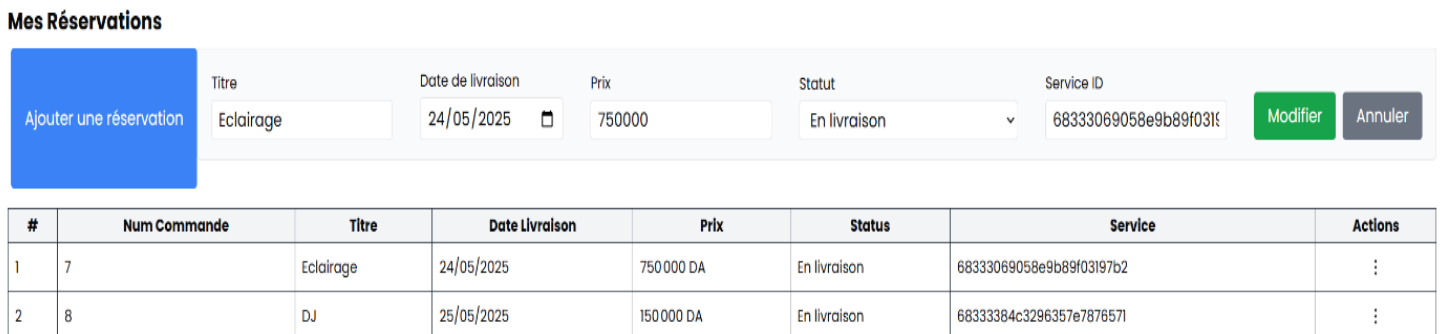


Figure 43: Interface Modifier une réservation

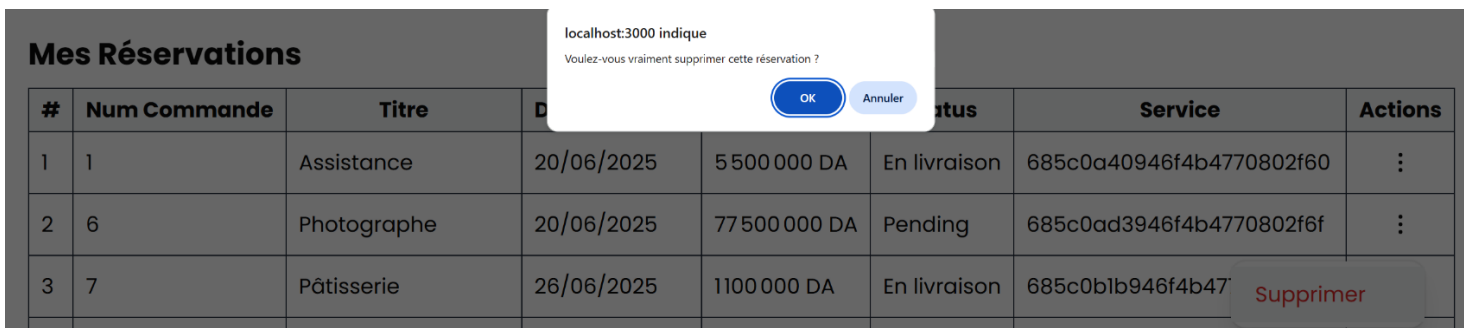


Figure 44: Interface Supprimer une réservation

### 3.11 Sprint 4

Ce sprint s'étend sur une période de deux semaines et inclut les activités suivantes :

- Ajouter un événement
- Modifier un événement
- Supprimer un événement

À l'issue de ce sprint, un incrément fonctionnel est produit, correspondant à la quatrième étape de développement de notre application. Nous présentons ci-après le diagramme des cas d'utilisation associé, les descriptions textuelles de ces cas, les diagrammes de séquence, ainsi que le diagramme de classes participantes.

Pour conclure, nous illustrons les interfaces utilisateurs réalisées durant ce sprint à travers des captures d'écran.

#### 3.11.1 Descriptions textuelles

Les tableaux 18, 19, 20 et 21 représentent respectivement les descriptions textuelles des cas d'utilisation ajouter, modifier, supprimer et annuler un évènement.

**Tableau 18 : Description textuelle du cas d'utilisation Ajouter un évènement**

<b>Titre</b>	Ajouter un événement
<b>Acteurs</b>	Administrateur
<b>Objectifs</b>	Organiser un nouvel événement basé sur des réservations faites par un client
<b>Pré conditions</b>	L'administrateur doit être authentifié Le client à qui est destiné l'évènement existe dans la BDD Le client a déjà fait au moins une réservation
<b>Scénario nominal</b>	1/ L'administrateur s'authentifie 2/ L'administrateur accède à l'interface Événements 3/ L'administrateur clique sur le bouton Ajouter un événement 4/ L'administrateur remplit les champs du formulaire d'ajout de l'évènement

	<p>5/ Cliquer sur le bouton Sauvegarder</p> <p>6/ Vérifier si le client existe dans la BDD</p> <p>7/Sauvegarder l'événement dans la BDD</p> <p>8/ Événement ajouté avec succès</p>
<b>Scénario alternatif</b>	<p>7/ Erreur le client n'existe pas</p> <p>Erreur lors de l'ajout de l'événement</p>
<b>Post conditions</b>	Événement ajouté avec succès

**Tableau 19 Description textuelle du cas d'utilisation Modifier un événement**

<b>Titre</b>	Modifier un événement
<b>Acteurs</b>	Administrateur
<b>Objectifs</b>	L'administrateur modifie le champ paiement à oui lorsque le client aura complété le paiement de l'événement
<b>Pré conditions</b>	<p>L'administrateur est déjà authentifié</p> <p>L'événement a déjà été créé</p>
<b>Scénario nominal</b>	<p>1/ L'administrateur s'authentifie</p> <p>2/ L'administrateur accède à l'interface Événements</p> <p>3/ L'administrateur choisit l'événement à modifier et clique sur le bouton Modifier dans la case Actions</p> <p>4/ L'administrateur change le champ paiement à oui et clique sur le bouton Sauvegarder</p> <p>5/ Événement modifié avec succès</p>
<b>Scénario alternatif</b>	/
<b>Post conditions</b>	Un événement a été modifié avec succès

**Tableau 20 : Description textuelle du cas d'utilisation Supprimer un événement**

<b>Titre</b>	Supprimer un événement
<b>Acteurs</b>	Administrateur

<b>Objectifs</b>	Supprimer un événement déjà existant (Supprimer l'événement de la BDD lorsqu'il sera terminé et libérer les services pris lors de cet événement)
<b>Pré conditions</b>	L'administrateur doit être authentifié L'événement doit être créé
<b>Scénario nominal</b>	1/ L'administrateur s'authentifie 2/ L'administrateur accède à l'interface Événements 3/ L'administrateur choisit l'événement à supprimer et clique sur le bouton Supprimer dans la case Actions 4/ Supprimer l'événement dans la BDD 5/ Événement supprimé avec succès
<b>Scénario alternatif</b>	/
<b>Post conditions</b>	Événement supprimé avec succès

**Tableau 21 : Description textuelle du cas d'utilisation Annuler un événement**

<b>Titre</b>	Annuler un événement
<b>Acteurs</b>	Administrateur
<b>Objectifs</b>	Supprimer un événement déjà existant (Par demande du client)
<b>Pré conditions</b>	L'administrateur doit être authentifié L'événement doit être créé
<b>Scénario nominal</b>	1/ L'administrateur s'authentifie 2/ L'administrateur accède à l'interface Événements 3/ L'administrateur choisit l'événement à annuler et clique sur le bouton Supprimer dans la case Actions 4/ Supprimer l'événement dans la BDD 5/ Événement annulé avec succès
<b>Scénario alternatif</b>	/
<b>Post conditions</b>	Événement annulé avec succès

	Les services réservés lors de cet événement sont libérés
--	--

### 3.11.2 Diagramme de cas d'utilisation du sprint 4

Dans la figure 45, on regroupe les informations des tableaux précédents qui décrivent les cas d'utilisation du sprint 4.

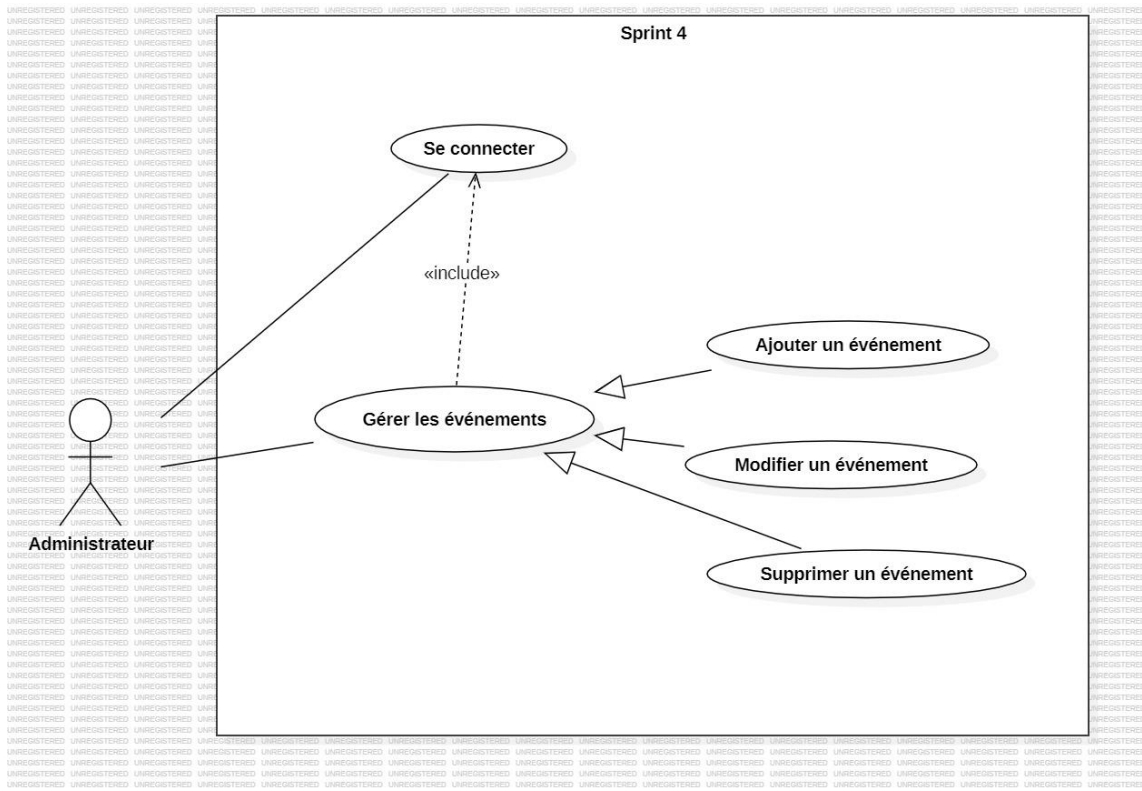


Figure 45: Diagramme de cas d'utilisation du sprint 4

### 3.11.3 Diagrammes de séquence

Les figures 46, 47, 48 et 49 représentent respectivement les diagrammes de séquence de (ajouter, modifier, supprimer et annuler un événement).

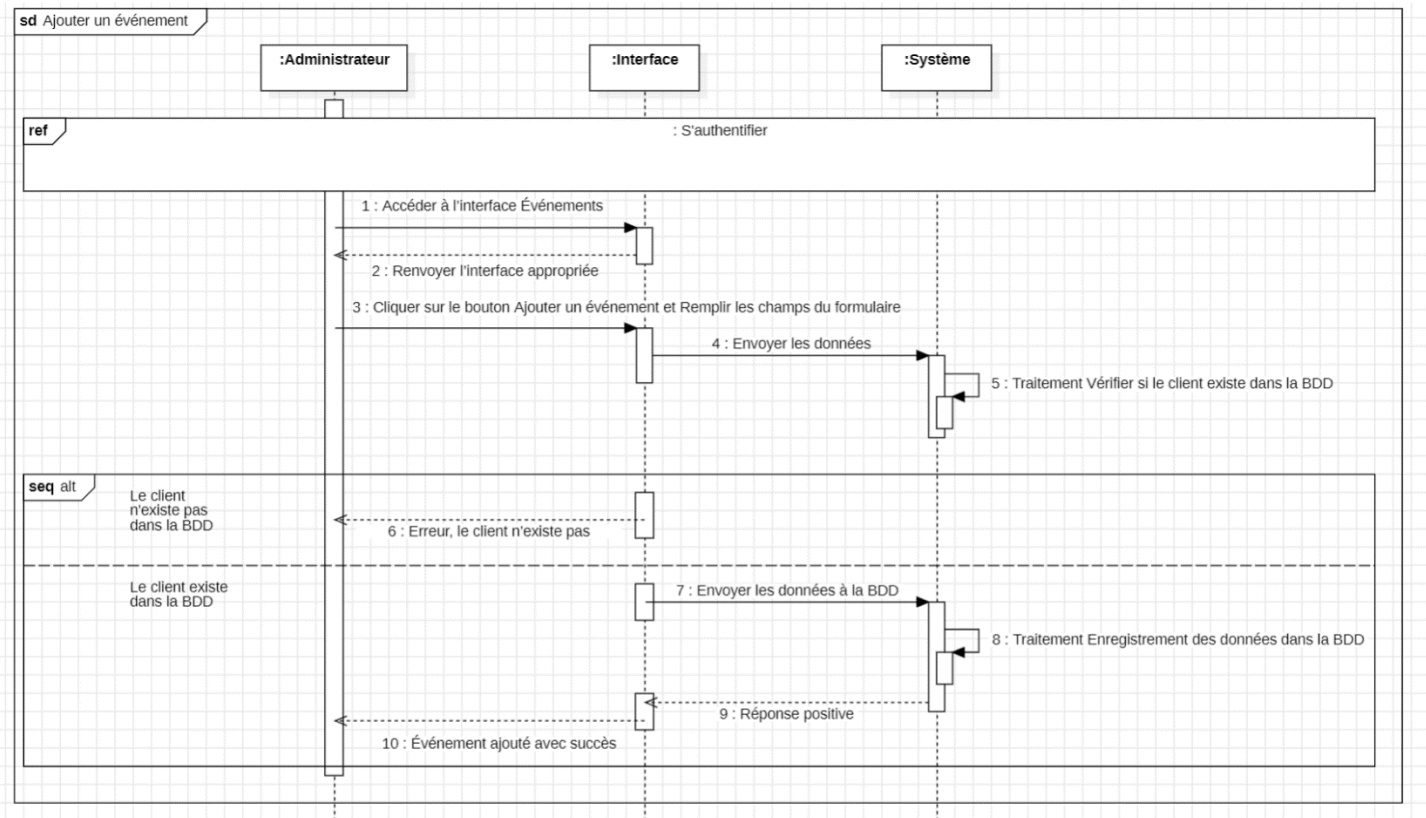


Figure 46: Diagramme de séquence du cas d'utilisation Ajouter un événement

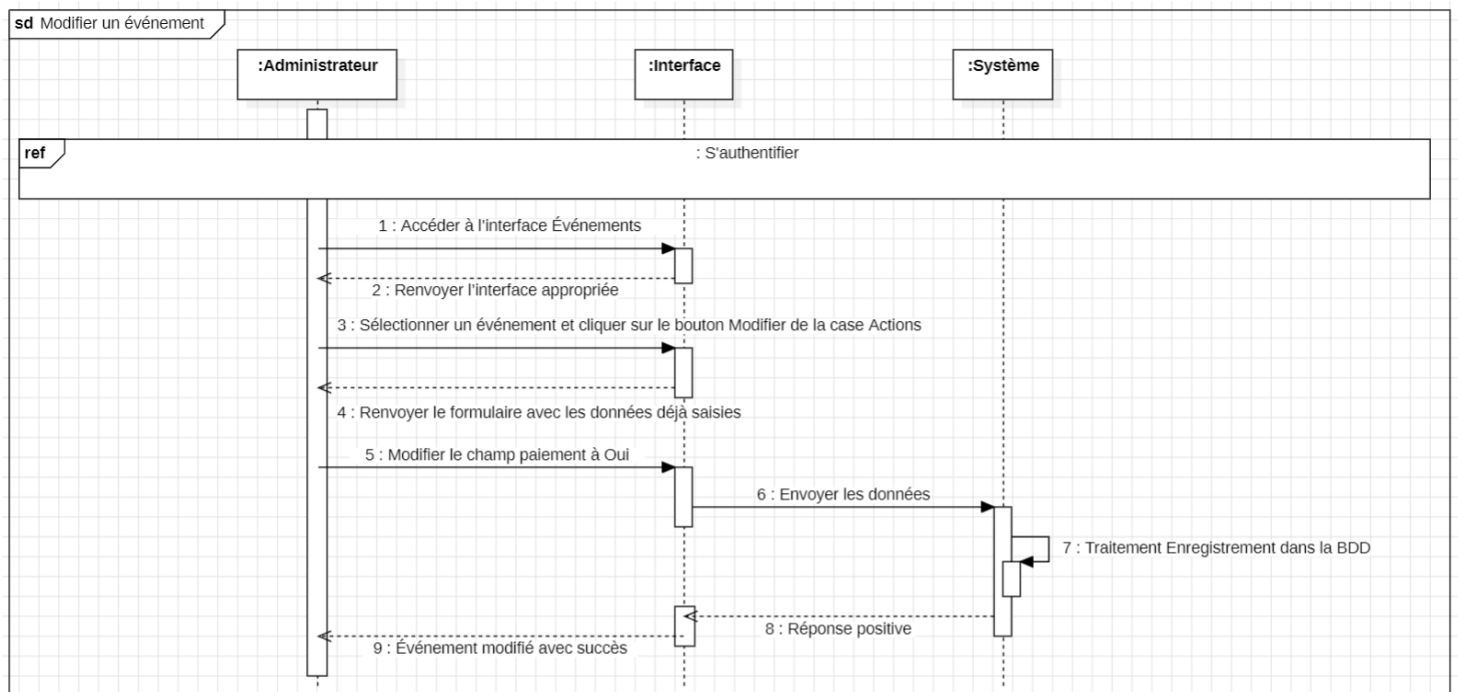
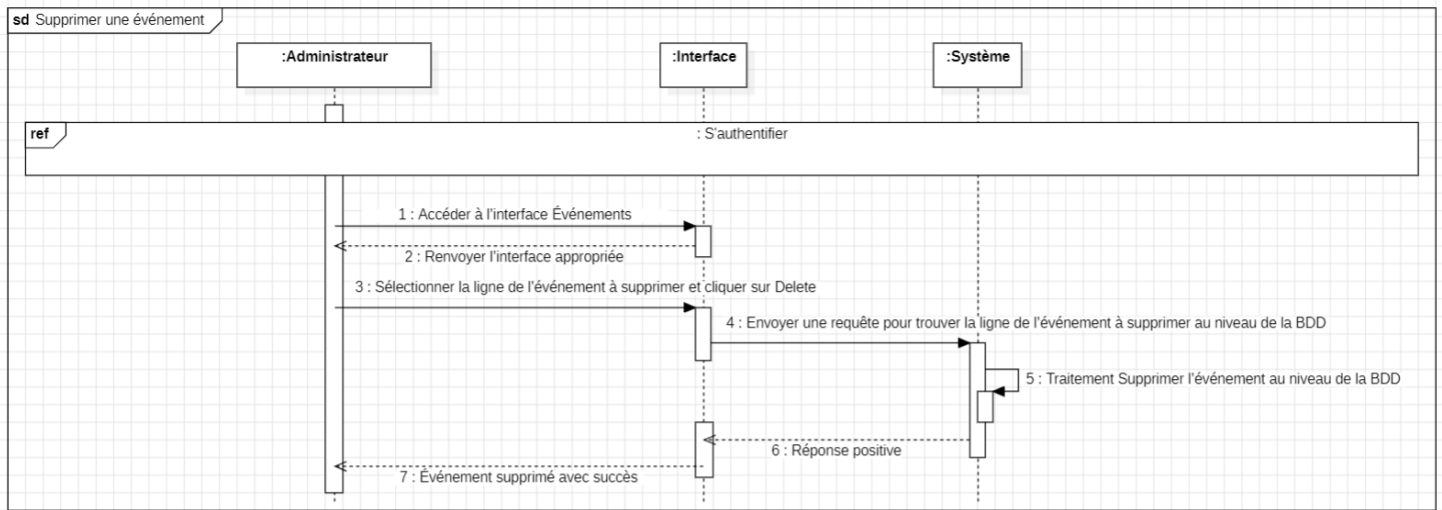
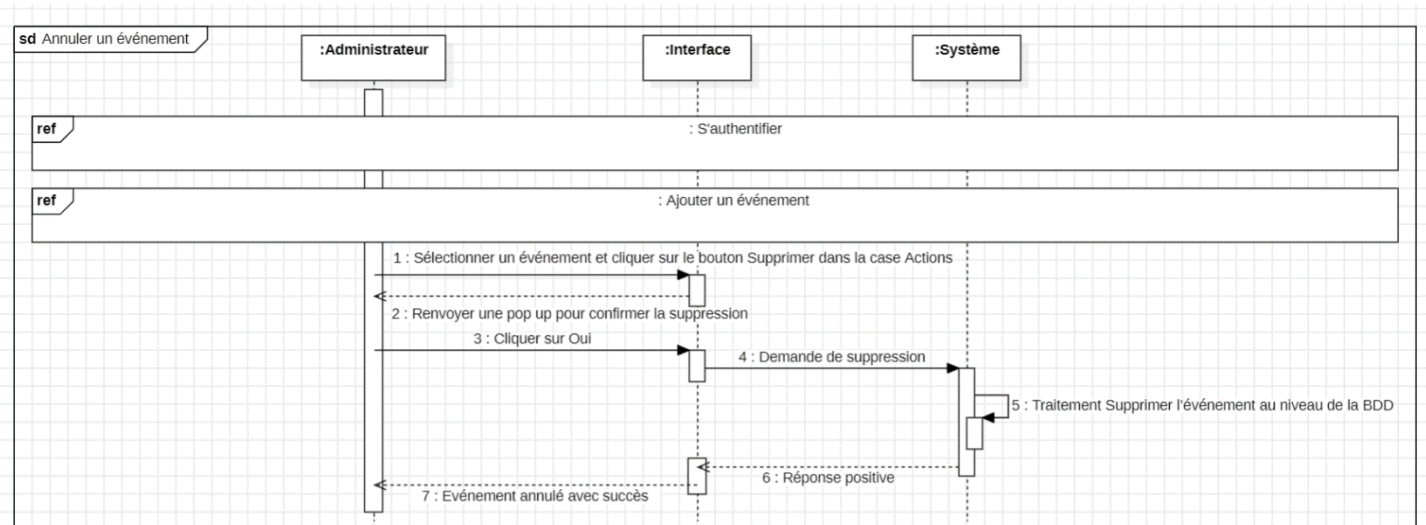


Figure 47: Diagramme de séquence du cas d'utilisation Modifier un événement



**Figure 48: Diagramme de séquence du cas d'utilisation Supprimer un événement**



**Figure 49: Diagramme de séquence du cas d'utilisation Annuler un événement**

### 3.11.4 Diagrammes de classes participantes

La figure 50 représente le diagramme de classes participantes du sprint 4

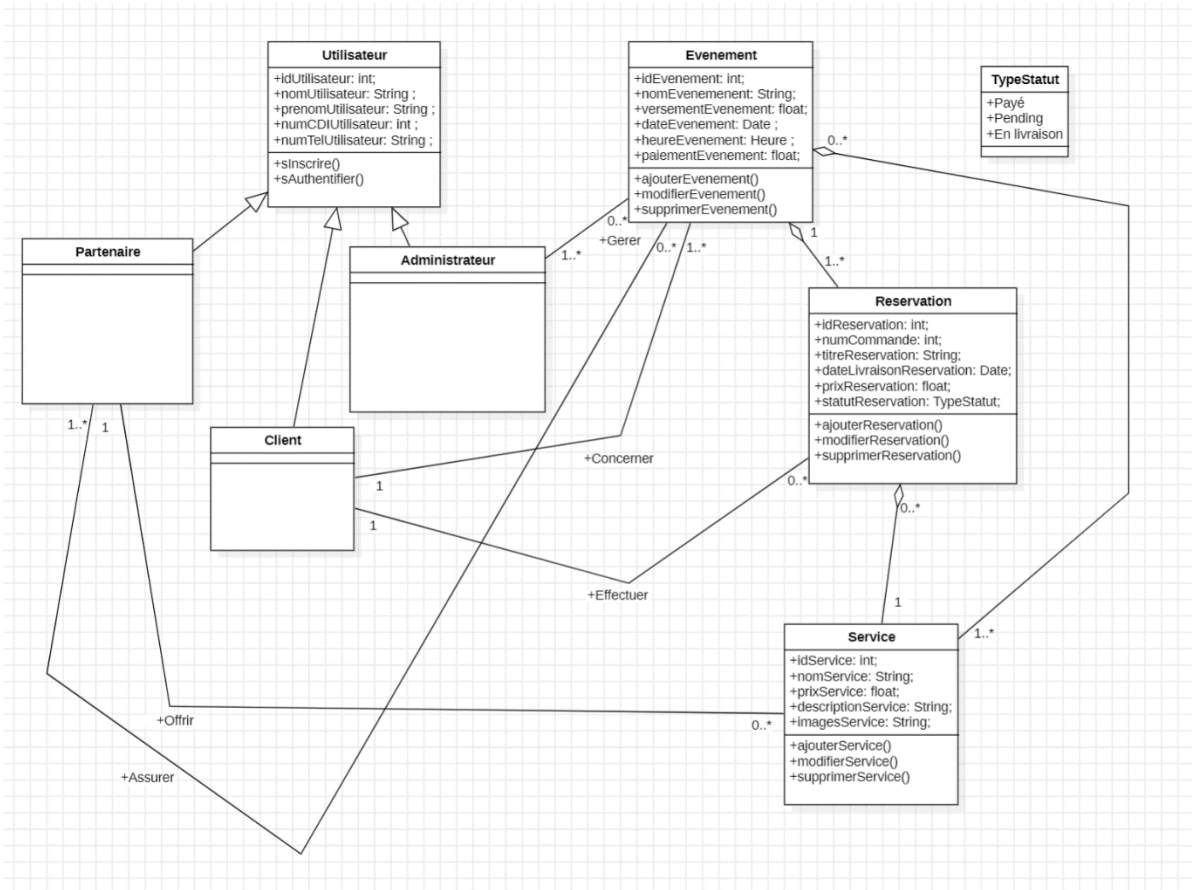


Figure 50: Diagramme de classes participantes du sprint 4

### 3.11.5 Interfaces utilisateur

Les figures 51, 52 et 53 représentent les interfaces d’ajout, modification, suppression et annulation d’un événement.

## Événements

Ajouter un événement

Titre	Client	Versement	Paiement	Actions
<input type="text" value="Choisir le titre de l'événement"/>	<input type="text" value="Numéro de carte d'identité"/>	<input type="text" value="% versement"/>	Non	<input type="button" value="Ajouter"/> <input type="button" value="Annuler"/>
Mariage	Inclientthree fnclientthree	30%	Non	<input type="button" value="⋮"/>
Fiançailles	Inclientfour fnclientfour	40%	Non	<input type="button" value="⋮"/>
Anniversaire	Inclientfive fnclientfive	50%	Oui	<input type="button" value="⋮"/>

Figure 51: Interface Ajouter un événement

# Événements

Ajouter un événement

Titre	Client	Versement	Paiement	Actions
Mariage	Inclientthree fnclientthree	30%	Oui ▾	Enregistrer Annuler
Fiançailles	Inclientfour fnclientfour	40%	Non	⋮
Anniversaire	Inclientfive fnclientfive	50%	Oui	⋮

Figure 52: Interface Modifier un événement

localhost:3001 indique  
Supprimer cet événement ?

OK Annuler

Espace Administrateur

## Événements

Ajouter un événement

Titre	Client	Versement	Paiement	Actions
Mariage	Inclientthree fnclientthree	30%	Non	⋮
Fiançailles	Inclientfour fnclientfour	40%	Non	⋮
Anniversaire	Inclientfive fnclientfive	50%	Oui	Voir détails réservations Modifier Supprimer

Figure 53: Interface Supprimer/Annuler un événement

### 3.12 Dictionnaire de données

Voici le dictionnaire de données :

**idUtilisateur:** Représente l'identifiant de l'utilisateur.

**nomUtilisateur:** Représente le nom de l'utilisateur.

**prenomUtilisateur:** Représente le prénom de l'utilisateur.

**numCDIUtilisateur:** Représente le numéro de la carte d'identité de l'utilisateur

**numTelUtilisateur:** Représente le numéro de téléphone de l'utilisateur.

**idCompte:** Représente l'identifiant du compte.

**emailCompte:** Représente l'email du compte.

**motDePasseCompte:** Représente le mot de passe du compte.

**idService:** Représente l'identifiant du service.

**nomService:** Représente le nom du service.

**prixService:** Représente le prix du service.

**descriptionService:** Représente la description du service.

**imagesService:** Représente les images du service.

**idReservation:** Représente l'identifiant de la réservation.

**titreReservation:** Représente le titre de la réservation.

**dateLivraisonReservation:** Représente la date de livraison de la réservation.

**prixReservation:** Représente le prix de la réservation.

**statutReservation:** Représente le statut de la réservation.

**idEvenement:** Représente l'identifiant de l'événement.

**nomEvenement:** Représente le nom de l'événement.

**versementEvenement:** Représente le versement de l'événement.

**dateEvenement:** Représente la date de l'événement.

**heureEvenement:** Représente l'heure de l'événement.

**paiementEvenement:** Représente le paiement de l'événement.

### 3.13 Diagramme de classes global :

La figure 54 représente le digramme de classes global de l'application (Gestion de fêtes et événements).

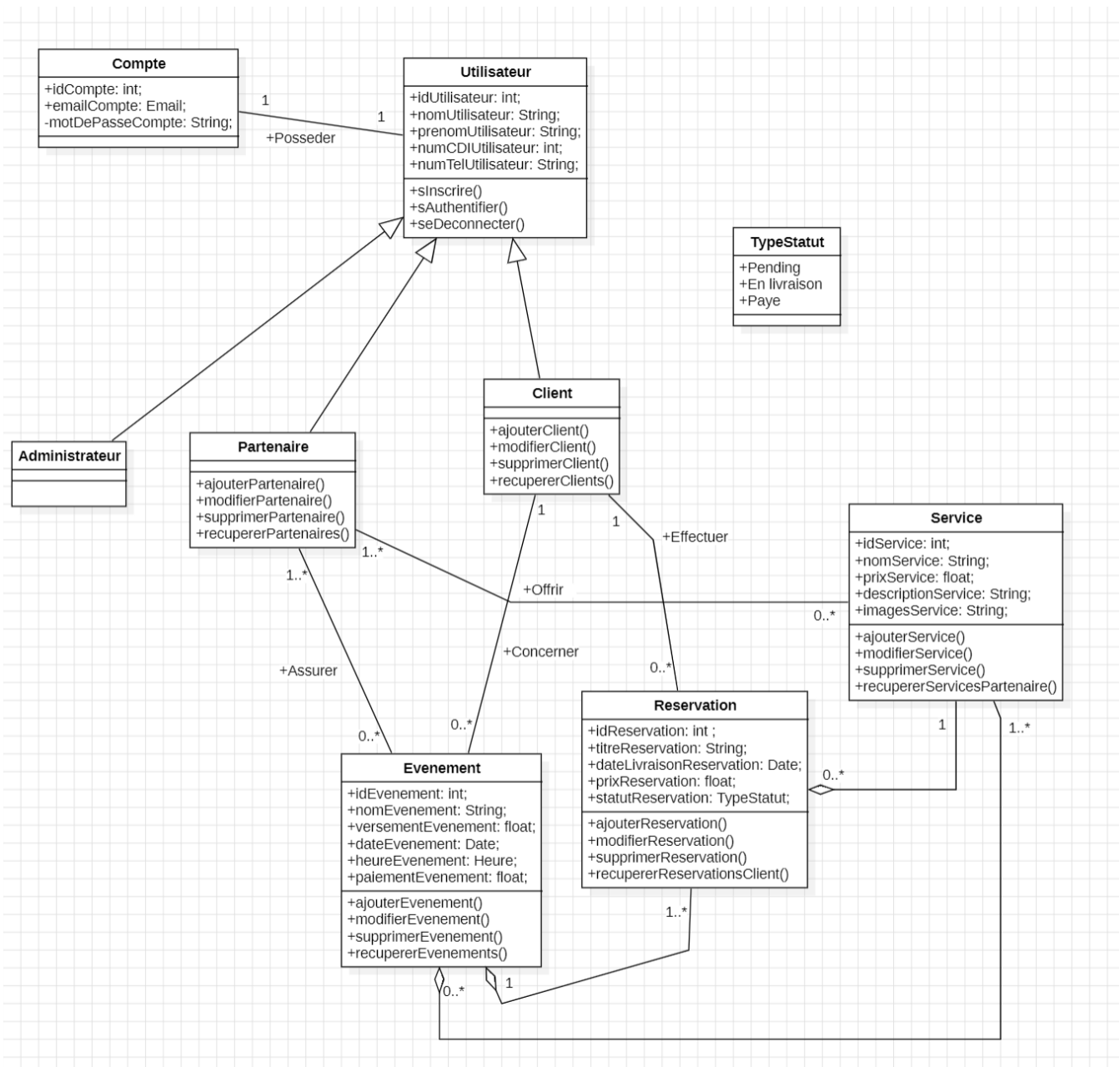


Figure 54: Diagramme de classes global

### **3.14 Conclusion**

La démarche Scrum appliquée dans ce chapitre a permis d'organiser efficacement le développement du système de gestion d'événements. Grâce à une planification par sprints, chaque fonctionnalité a été pensée, modélisée et implémentée de manière progressive. L'identification des acteurs, la définition claire des user stories, ainsi que l'usage de diagrammes et d'interfaces graphiques ont contribué à assurer la cohérence et la pertinence des choix techniques et fonctionnels. Cette approche itérative a favorisé une meilleure adaptation aux besoins réels des utilisateurs, tout en garantissant une traçabilité claire de l'évolution du projet.

## Conclusion Générale

---

Au terme de ce stage, nous avons eu l'opportunité de concevoir et de développer une application web complète dédiée à l'organisation d'événements privés et professionnels. Ce projet, mené de bout en bout selon la méthodologie Scrum, m'a permis d'appliquer une approche agile, structurée en sprints, favorisant une gestion efficace des tâches, une amélioration continue et une forte réactivité face aux besoins fonctionnels.

Côté technique, j'ai mobilisé un ensemble de technologies modernes et complémentaires : **HTML5**, **CSS3** et **JavaScript** pour la structure et le style, **React** et **Next.js** pour un frontend interactif et performant, et **Node.js** avec **Express.js** pour construire une API robuste côté backend. L'ensemble des données est géré via **MongoDB**, garantissant une bonne flexibilité et scalabilité du système.

L'application finale est pleinement fonctionnelle, conforme au cahier des charges initial, et intègre toutes les principales fonctionnalités attendues : gestion des services, des réservations, des événements, et des utilisateurs. Ce stage m'a permis non seulement de renforcer mes compétences techniques, mais aussi de développer une réelle autonomie, un esprit de rigueur, ainsi qu'une capacité à travailler dans un cadre méthodologique professionnel.

## Bibliographie

BigNova. (s.d.). Récupéré sur <https://bignova-company.com/>

Cantelon, M. H. (2014). *Node.js in Action*. Manning Publications. (2nd ed.). .

Duckett, J. (2011). *HTML and CSS: Design and Build Websites*. . Wiley.

Grinberg, F. .. (2018). *Full-Stack React Projects*. Packt Publishing.

Holtz, E. .. (2021). *Express.js Guide: The Comprehensive Book on Express.js*. . Leanpub.

Media., O. (2020). Banks, A., & Porcello, E.. *Learning React* . (3rd ed.).

MongoDB, Inc. *MongoDB Documentation*. <https://www.mongodb.com/docs/>. (2024).

Mozilla Developer Network (MDN). . *HTML et CSS documentation*. <https://developer.mozilla.org/fr/>. (2024).

Schwaber, K. &. (2020). *The Scrum Guide – The Definitive Guide to Scrum: The Rules of the Game*.  
*Scrum.org*. <https://scrumguides.org/>.

*Scrum.org*. *The Home of Scrum*. <https://www.scrum.org/>. (2024).

W3Schools. (2024). *HTML, CSS, JavaScript Tutorials*. <https://www.w3schools.com/>.

**Résumé :** Ce mémoire porte sur la conception et le développement d'une solution numérique visant à améliorer l'organisation d'événements. L'analyse des pratiques actuelles a révélé plusieurs limites majeures, telles que le manque de coordination entre prestataires et l'absence de planification centralisée, etc. Face à ces constats, une réponse technologique a été envisagée sous la forme d'une application intuitive, centralisée et accessible, capable de répondre efficacement aux besoins des utilisateurs, qu'ils soient particuliers ou professionnels. Pour structurer et encadrer la réalisation du projet, la méthode Scrum a été utilisée, permettant une progression par étapes, en tenant compte des priorités fonctionnelles et des retours au fil du développement. L'architecture logicielle repose sur le modèle MVC, accompagnée du langage de modélisation UML et de technologies modernes comme JavaScript et Next.js, offrant une base solide et évolutive. Le cœur du mémoire est consacré à la mise en œuvre du projet à travers une série de sprints, illustrant l'évolution de l'application. Chaque phase comprend la définition des besoins (user stories), la modélisation technique (cas d'utilisation, diagrammes de séquence et de classes), ainsi que le développement des interfaces graphiques. Cette démarche structurée a permis d'aboutir à un produit final cohérent, fonctionnel et parfaitement adapté aux besoins identifiés, démontrant l'efficacité d'une organisation méthodique dans le cadre d'un projet informatique.

**Mots clés :** Application, web, Scrum, sprints, UML, MVC, Javascript, NextJ, technologie.

**Abstract :** This master's dissertation focuses on the design and development of a digital solution aimed at improving event organization. The analysis of current practices revealed several major limitations, such as the lack of coordination between service providers and the absence of centralized planning, etc. In response, a technological solution was envisioned in the form of an intuitive, centralized, and accessible application that effectively meets the needs of both individual and professional users. To structure and guide the project development, the Scrum methodology was applied, allowing progress in iterative phases while considering functional priorities and feedback throughout the process. The software architecture is based on the MVC model, supported by the UML modeling language and modern technologies such as JavaScript and Next.js, providing a solid and scalable foundation. The core of this dissertation is dedicated to the practical implementation of the project, presented through a series of sprints that illustrate the application's evolution. Each phase includes the definition of needs (user stories), technical modeling (use case, sequence, and class diagrams), and the development of user interfaces. This structured approach resulted in a coherent, functional product that is well aligned with the identified needs, demonstrating the effectiveness of a methodical organization in the context of a software development project.

**Keywords :** Application, web, Scrum, UML, MVC, Javascript, NextJ, technology.