

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche  
Scientifique

Université Abderrahmane Mira De Bejaïa

Faculté des Sciences Exactes

Département d'Informatique



Mémoire de fin de cycle en vue de l'obtention du Master  
professionnel en informatique

Option : Génie Logiciel

---

# Conception et réalisation d'une plateforme de garde d'enfants à domicile

---

*Réalisé par :*

Mlle. HOCINI Esma

Mlle. TOUATI Meriem Yasmine

*Soutenu le 02 Juillet 2023*

*Devant le jury composé de :*

M. ACHROUFENE Achour - Président (Univ. A/Mira Bejaia)  
Mme. EL BOUHISSI Houda - Examineur (Univ. A/Mira Bejaia)  
M ALLEM Khaled - Encadrant (Univ. A/Mira Bejaia)

Promotion : 2022/2023

# Dédicace

“

*À nos familles, qui ont toujours été nos plus grands  
encourageants et soutiens.*

*À nos amis proches, nous souhaitons exprimer notre  
reconnaissance pour leur présence dans nos vies.*

*À nous-mêmes, en reconnaissance de notre travail  
acharné, de notre persévérance et de notre détermination à  
atteindre nos objectifs académiques.*

”

*- [Esma et Meriem Yasmine].*

# Remerciements

Nous tenons tout d'abord à rendre Grâce à Dieu tout Puissant de nous avoir donné la patience et la volonté nécessaires à la réalisation de cet humble travail.

Nous sommes reconnaissantes envers M. A.ACHROUFENE et Mme. H.El BOUHISSI membres du jury pour leurs conseils avisés, nous sommes honorées d'avoir pu bénéficier de votre évaluation et de votre claire-voyance.

Nous souhaitons remercier M. K.ALLEM, pour son encadrement, ses conseils et éclaircissements. Votre engagement a été essentiel à la réussite de ce projet.

Nous tenons également à remercier les membres de l'entreprise d'accueil BigNova pour leurs précieuses contributions, leurs aides, leurs suggestions, et les commentaires constructifs.

L'élaboration de ce mémoire a été une expérience captivante, et nous sommes sincèrement reconnaissantes envers tous ceux qui ont joué un rôle de près ou de loin dans sa concrétisation.

Merci à tous.

# Résumé

Face à la difficulté pour de nombreux parents de trouver des places en crèche pour leurs enfants, il devient crucial de proposer des alternatives de confiance. Dans cette optique, notre travail avait pour objectif de développer une plate-forme en ligne permettant aux parents de trouver et de contacter des gardiens d'enfants à domicile dignes de confiance. Cette plate-forme se positionne comme une solution pratique pour les parents à la recherche de services de garde d'enfants personnalisés. Pour la mise en œuvre, nous avons suivi la méthode agile SCRUM pour la gestion de projet combinée au processus unifié UP ainsi que l'utilisation d'une panoplie de bibliothèques et de frameworks de développement.

---

**Mots clés :** Garde d'enfants, UML, Scrum, Processus unifié, React Js, Nest Js.

---

# Abstract

Given the difficulty many parents have in finding daycare places for their children, it's becoming crucial to offer trusted alternatives. With this in mind, the aim of our work was to develop an online platform enabling parents to find and contact trustworthy home-based childminders. This platform is positioned as a practical solution for parents looking for personalized childcare services.

In short, this work highlights the efforts being made to bridge the imbalance between the supply of crèches and the demand from parents in Algeria by connecting parents with qualified, trustworthy professionals. For the implementation, we followed the SCRUM agile method for project management combined with the UP unified process as well as the use of a set of development libraries and frameworks..

---

**Keywords :** Babysitting, Nannies, UML, Scrum, Unified Process, React Js, Nest Js.

---

# Table des matières

Table des matières . . . . .	I
Liste des figures . . . . .	IV
Liste des tableaux . . . . .	VI
Liste des acronymes . . . . .	VII
<b>Introduction générale . . . . .</b>	<b>1</b>
<b>1 Présentation de l'organisme d'accueil et étude de l'existant . . . . .</b>	<b>3</b>
1.1 Introduction . . . . .	4
1.2 Présentation de l'entreprise d'accueil . . . . .	4
1.2.1 Description de l'entreprise . . . . .	4
1.2.2 Les différents clients et partenaires . . . . .	5
1.2.3 Les services proposés par l'entreprise . . . . .	5
1.3 Présentation du domaine métier . . . . .	6
1.3.1 Les exigences du domaine . . . . .	6
1.3.2 Intérêts du domaine . . . . .	6
1.3.3 Les types de garde d'enfants à domicile . . . . .	7
1.3.4 Recrutement . . . . .	8
1.4 Quelques sites de garde de référence . . . . .	8
1.4.1 Au niveau national . . . . .	8
1.4.2 Au niveau international . . . . .	8
1.5 Problématique . . . . .	10
1.6 Solution envisagée . . . . .	11
1.7 Conclusion . . . . .	11
<b>2 Langages de modélisation et méthodologies de développement . . . . .</b>	<b>12</b>
2.1 Introduction . . . . .	13
2.2 Langages de modélisation . . . . .	13
2.2.1 Définition d'UML . . . . .	13
2.2.2 Définition d'OCL . . . . .	13
2.2.3 Diagrammes utilisés . . . . .	14
2.3 Méthodologie de développement . . . . .	14
2.3.1 SCRUM . . . . .	14
2.3.2 Processus unifié (UP) . . . . .	16
2.3.3 Combinaison des deux méthodes . . . . .	16
2.4 Conclusion . . . . .	17
<b>3 Sprint zéro . . . . .</b>	<b>18</b>
3.1 Introduction . . . . .	19

3.2	Expression des besoins . . . . .	19
3.2.1	Identification des acteurs . . . . .	19
3.2.2	Diagramme de contexte . . . . .	19
3.2.3	Équipe Scrum . . . . .	21
3.2.4	Définition des besoins fonctionnels . . . . .	22
3.3	Planification avec Scrum . . . . .	25
3.3.1	Création du Backlog de produit . . . . .	25
3.3.2	Planification des sprints . . . . .	27
3.4	Présentation du logo de la plate-forme . . . . .	27
3.5	Configuration de l’environnement de développement . . . . .	27
3.5.1	Librairies utilisées . . . . .	28
3.5.2	Environnement de développement . . . . .	28
3.5.3	Quelques outils d’organisation et de planification . . . . .	29
3.5.4	Outils de test et de documentation . . . . .	29
3.6	Conclusion . . . . .	30
<b>4</b>	<b>Sprint un . . . . .</b>	<b>31</b>
4.1	Introduction . . . . .	32
4.2	Backlog du Sprint 1 . . . . .	32
4.3	Spécifications . . . . .	33
4.3.1	Diagrammes de cas d’utilisation . . . . .	34
4.3.2	Descriptions textuelles . . . . .	34
4.4	Analyse . . . . .	39
4.4.1	Modèle de domaine . . . . .	39
4.4.2	Diagrammes de séquence système . . . . .	40
4.5	Conception . . . . .	44
4.5.1	Diagrammes de séquences détaillés . . . . .	44
4.5.2	Diagrammes d’activité . . . . .	45
4.5.3	Diagramme des classes de conception . . . . .	46
4.6	Implémentation . . . . .	47
4.6.1	Modèle relationnel de données . . . . .	48
4.7	Présentation d’interfaces . . . . .	48
4.8	Conclusion . . . . .	52
<b>5</b>	<b>Sprint deux . . . . .</b>	<b>53</b>
5.1	Introduction . . . . .	54
5.2	Backlog du Sprint 2 . . . . .	54
5.3	Spécifications . . . . .	55
5.3.1	Diagrammes de cas d’utilisation . . . . .	55
5.3.2	Descriptions textuelles . . . . .	55
5.4	Analyse . . . . .	57
5.4.1	Modèle de domaine . . . . .	57
5.4.2	Diagrammes de séquence système . . . . .	58
5.5	Conception . . . . .	61
5.5.1	Diagrammes de séquences détaillés . . . . .	61
5.5.2	Diagrammes de classe de conception . . . . .	62

5.6	Implémentation . . . . .	63
5.6.1	Modèle relationnel de données . . . . .	64
5.7	Présentation d'interfaces . . . . .	64
5.8	Conclusion . . . . .	66
<b>6</b>	<b>Sprint trois . . . . .</b>	<b>67</b>
6.1	Introduction . . . . .	68
6.2	Backlog du Sprint 3 . . . . .	68
6.3	Spécification . . . . .	69
6.3.1	Diagrammes des cas d'utilisation . . . . .	69
6.3.2	Descriptions textuelles . . . . .	71
6.4	Analyse . . . . .	74
6.4.1	Modèle de domaine . . . . .	75
6.4.2	Diagrammes de séquence système . . . . .	76
6.5	Conception . . . . .	79
6.5.1	Diagrammes de séquences détaillés . . . . .	79
6.5.2	Diagrammes d'activité . . . . .	83
6.5.3	Diagramme de classe de conception . . . . .	86
6.5.4	Diagramme de classes global . . . . .	87
6.6	Implémentation . . . . .	88
6.6.1	Modèle relationnel de données . . . . .	88
6.7	Présentation d'interfaces . . . . .	89
6.7.1	Interface de gestion des offres de gardes . . . . .	89
6.8	Conclusion . . . . .	92
	<b>Conclusion et perspectives . . . . .</b>	<b>93</b>
	<b>Annexes . . . . .</b>	<b>96</b>
<b>A</b>	<b>Quelques technologies utilisées . . . . .</b>	<b>97</b>
<b>B</b>	<b>Diagrammes de séquence système . . . . .</b>	<b>100</b>
<b>C</b>	<b>Diagramme de Gantt des Sprints . . . . .</b>	<b>103</b>
<b>D</b>	<b>Descriptifs des réunions effectuées . . . . .</b>	<b>104</b>
<b>E</b>	<b>Questionnaire pour les crèches . . . . .</b>	<b>106</b>



# Liste des figures

1.1	Logo de BigNova. . . . .	4
1.2	Types de garde d'enfants à domicile. . . . .	8
2.1	Schéma explicatif du fonctionnement de Scrum . . . . .	16
3.1	Diagramme de contexte dynamique. . . . .	20
3.2	Constitution de l'équipe Scrum. . . . .	21
3.3	Diagramme de cas d'utilisation global. . . . .	24
3.4	Logo de KODOMO . . . . .	27
3.5	Logo de ReactJS. . . . .	28
3.6	Logo de NestJS. . . . .	28
3.7	Logo de VSC. . . . .	28
4.1	Diagramme de cas d'utilisation du Sprint 1 . . . . .	34
4.2	Modèle de domaine du sprint 1. . . . .	40
4.3	Diagramme de séquence système du cas d'utilisation «s'authentifier». . . . .	41
4.4	Diagramme de séquence système du cas d'utilisation «administrer un gestionnaire». . . . .	42
4.5	Diagramme de séquence système du cas d'utilisation «traiter le profile d'un gardien». . . . .	43
4.6	Diagramme de séquence système du cas d'utilisation «s'authentifier». . . . .	45
4.7	Diagramme d'activité du cas d'utilisation «s'authentifier». . . . .	46
4.8	Diagramme de classes de conception du sprint 1. . . . .	47
4.9	Interfaces de connexion du back office. . . . .	49
4.10	Interface de gestion des gardiens. . . . .	50
4.11	Interface d'ajout (1). . . . .	50
4.12	Interface du profil d'un gardien. . . . .	51
5.1	Diagramme de cas d'utilisation du Sprint 2. . . . .	55
5.2	Modèle de domaine du sprint 2. . . . .	58
5.3	Diagramme de séquence système du cas d'utilisation «administrer un gestionnaire». . . . .	59
5.4	Diagramme de séquence système du cas d'utilisation «traiter le profile d'un gardien». . . . .	60
5.5	Diagramme de séquence système du cas d'utilisation «administrer un gestionnaire». . . . .	62
5.6	Diagramme de classe de conception du sprint 2. . . . .	63
5.7	Section de gestion tous les admins. . . . .	64
5.8	Interface des gestionnaires et administrateurs. . . . .	65

5.9	Interface du tableau de bord (1).	66
5.10	Interface du tableau de bord sombre(2).	66
6.1	Diagramme de cas d'utilisation du Sprint 3.	70
6.2	Modèle de domaine du sprint 3	75
6.3	Diagramme de séquence système du cas d'utilisation «créer une offre de garde».	76
6.4	Diagramme de séquence système du cas d'utilisation «postuler à une offre de garde».	77
6.5	Diagramme de séquence système du cas d'utilisation «Gérer un entretien».	78
6.6	Diagramme de séquence détaillé du cas d'utilisation «créer une offre de garde».	80
6.7	Diagramme de séquence détaillé du cas d'utilisation «postuler à une offre de garde».	81
6.8	Diagramme de séquence détaillé du cas d'utilisation «Gérer un entretien».	82
6.9	Diagramme d'activité du cas d'utilisation «créer une offre de garde».	83
6.10	Diagramme d'activité du cas d'utilisation «gérer un rendez-vous».	85
6.11	Diagramme de classes conception du sprint 3.	86
6.12	Diagramme des classes global.	87
6.13	Interface de gestion des offres de garde.	89
6.14	Interface de gestion des offres de garde.(suite)	90
6.15	Liste des rendez-vous	91
B.1	Diagramme de séquence système du cas d'utilisation «créer un compte».	101
B.2	Diagramme de séquence système du cas d'utilisation «réinitialisation de mot de passe».	102

# Liste des tableaux

1.1	Tableau comparatif de quelques sites de gardes d'enfants. . . . .	9
2.1	Avantages des deux méthodologies. . . . .	17
3.1	Tableau de messages échangés. . . . .	20
3.2	Le Backlog du produit général. . . . .	26
3.3	Tableau des dates du début et fin de chaque sprint . . . . .	27
4.1	Backlog du Sprint 1. . . . .	32
4.3	Règles de gestion : S'authentifier. . . . .	35
4.4	Description textuelle du CU : S'authentifier. . . . .	35
4.5	Description textuelle du CU : Gérer le compte d'un client. . . . .	36
4.7	Description textuelle du CU : Traiter le profil d'un gardien. . . . .	37
4.9	Règles de gestion : Bannir un gardien. . . . .	38
4.10	Description textuelle du CU : Bloquer le compte d'un client. . . . .	38
5.1	Backlog du Sprint 2. . . . .	54
5.3	Documentation CU : Administrer un gestionnaire. . . . .	56
5.5	Documentation CU : Consulter les statistiques du système. . . . .	57
6.1	Backlog du Sprint 3. . . . .	68
6.2	Règles de gestion : Créer une offre de garde. . . . .	71
6.3	Documentation CU : Créer une offre de garde. . . . .	71
6.4	Règles de gestion : Gérer un entretien entre un parent et un gardien. . . . .	72
6.5	Documentation CU : Gérer un entretien entre un parent et un gardien. . . . .	72
6.6	Règles de gestion : Postuler à une offre de garde. . . . .	73
6.7	Documentation CU : Postuler à une offre de garde. . . . .	73
6.8	Documentation CU : Créer une tâche de garde. . . . .	74
C.1	Diagramme de Gantt des Sprints. . . . .	103
D.1	Tableau des réunions. . . . .	105

# Liste des sigles et acronymes

<b>API</b>	<i>Application Programming Interface</i>
<b>URL</b>	<i>Uniform Resource Locator</i>
<b>CRUD</b>	<i>Create Read Update Delete</i>
<b>HTTP</b>	<i>Hypertext Transfer Protocol</i>
<b>JSON</b>	<i>JavaScript Object Notation</i>
<b>NPM</b>	<i>Node Package Manager</i>
<b>REST</b>	<i>Representational State Transfer</i>
<b>SEO</b>	<i>Search Engine Optimization</i>
<b>UML</b>	<i>Unified Modeling Language</i>
<b>UP</b>	<i>Unified Process</i>

# Introduction générale

Dans de nombreux pays à travers le monde, les parents, en particulier ceux qui sont employés à temps plein ou qui ont d'importantes responsabilités professionnelles, font face au besoin de faire garder leurs enfants à domicile, communément appelé le babysitting. On remarque qu'en Algérie, il existe un déséquilibre entre l'offre de crèches et la demande des parents en matière de garderie, et cela complique davantage la tâche des parents pour trouver des professionnels compétents et fiables pour prendre soin de leurs enfants.

De nos jours l'outil informatique joue un rôle crucial dans toute société, transformant de nombreux aspects de notre vie contemporaine en facilitant l'exécution de diverses tâches. L'objectif principal de notre travail est de développer une plate-forme qui met en relation les parents avec des personnes compétentes, passionnées et investies par le métier de gardien d'enfants.

En travaillant en étroite collaboration avec les différents membres et responsables du projet et en appliquant les principes de Scrum, nous avons organisé le travail en itérations appelées "sprints", dans lesquelles on aura défini des objectifs spécifiques et livré des parties du produit.

En menant ce projet nous avons structuré son élaboration en six chapitres.

Le premier, intitulé "Présentation de l'organisme d'accueil et étude de l'existant", est consacré à la présentation de l'entreprise d'accueil, BigNova. Nous y examinons les principaux enjeux et défis qui se présentent dans le domaine de garde d'enfants en Algérie.

Dans le deuxième chapitre, nous introduirons la méthode de développement utilisée dans le cadre de notre projet. Nous justifierons ensuite le choix d'utiliser une méthodologie de développement mixte, qui combine les principes de Scrum et ceux du Processus unifié (UP).

Le troisième chapitre est dédié au Sprint Zéro, une phase d'amorçage du projet où nous aurons défini le périmètre et les objectifs de ce dernier. Nous détaillerons la constitution de l'équipe, la définition des tâches, ainsi que la gestion des risques et des responsabilités.

Les quatrième, cinquième et sixième chapitres couvriront les sprints réalisés pour le développement de l'application. Dans le quatrième chapitre, nous décrirons en détail les réalisations du Sprint 1. Par la suite nous nous attacherons dans le cinquième chapitre à effectuer toutes les fonctions du Sprint 2. Puis, dans le sixième chapitre, nous nous concentrerons sur le Sprint 3 et les fonctionnalités majeures implémentées.

En conclusion on terminera ce travail par une récapitulation générale de l'objet de ce mémoire et les perspectives en vue de son amélioration.

# Chapitre 1

## Présentation de l'organisme d'accueil et étude de l'existant

### 1.1 Introduction

Dans ce chapitre, nous commencerons par introduire l'entreprise d'accueil BigNova, les services qu'elle propose et quelques unes des ces réalisations. Nous poursuivons ensuite notre démarche principalement par un examen approfondi des aspects actuels et des défis rencontrés dans le domaine de la petite enfance, en mettant spécifiquement l'accent sur la prospection des failles des crèches, qui constituent le cœur de notre problématique. A la fin, nous proposerons une solution adaptée en mettant en avant les avantages offerts par la garde à domicile comme alternative pour les parents qui font face à des difficultés avec les crèches.

### 1.2 Présentation de l'entreprise d'accueil

*BigNova* est une entreprise de développement informatique qui se consacre à fournir des solutions technologiques innovantes pour tout établissement allant des start-ups aux grandes sociétés. BigNova a rapidement évolué dans l'industrie grâce à son expertise technique, sa passion pour l'innovation et son engagement envers la satisfaction de ses clients. Son principal objectif est d'aider les entreprises à relever les défis technologiques auxquels elles sont confrontées en leur proposant des solutions personnalisées qui répondent au mieux à leurs besoins. Que ce soit pour la création de sites web, d'applications mobiles, ou d'outils d'intelligence artificielle afin d'améliorer l'expérience utilisateur (voir la section 1.2.2 pour le détail des références sur les réalisations de Big Nova). Ci-dessous 1.1 le logo de l'entreprise.



FIG. 1.1 : Logo de BigNova.

#### 1.2.1 Description de l'entreprise

L'équipe de direction de Big Nova est composée de professionnels qualifiés et habiles dans le domaine du développement créatif et réactif. Ils ont une vision claire et novatrice pour l'entreprise, ainsi qu'une passion pour la résolution des problèmes auxquels peuvent faire face leurs clients, ce faisant ils peuvent offrir des solutions technologiques innovantes. En plus de l'équipe de direction, l'entreprise dispose également d'un personnel clé, confirmé dans divers domaines, notamment la programmation, la conception, l'optimisation digitale et le service client.

Tous les membres de l'équipe sont expérimentés, passionnés par leurs travaux, et sont

déterminés à fournir à chaque client le meilleur de leurs expertises. L'équipe de direction ainsi que son personnel clé constituent un des atouts majeurs de l'entreprise et sont essentiels à l'accomplissement de son succès.

### 1.2.2 Les différents clients et partenaires

- Quelques clients répertoriés chez l'entreprise :
  - Live Resto <sup>1</sup> est un site de livraison de plats à domicile ou au bureau, essentiellement sur Marseille.
  - NBK <sup>2</sup> est un groupe de consulting qui se focalise sur les services financiers, le développement commercial, les solutions numériques, ainsi que les conseils juridiques en Fintech et en néobanque. <sup>3</sup>
  - LISL UP <sup>4</sup> est une application mobile d'écologie de soi basée sur une approche holistique et scientifique.
- Les Partenaires : Infoware<sup>5</sup>, une entreprise spécialisée dans la gestion et la résolution des problèmes informatiques.

### 1.2.3 Les services proposés par l'entreprise

Big Nova fournit un lot de services diversifiés, principalement :

- La création de sites web et d'applications mobiles.
- La conception des systèmes de vision par ordinateur (Intelligence artificielle).
- La conception de logo, affiches publicitaires, éléments 3D.
- Le Web design.
- L'Aide au référencement et optimisation des sites web SEO.

L'entreprise s'est intéressée au domaine de la petite enfance dans ce qui suit nous présentons quelques aspects de ce domaine d'activité.

---

<sup>1</sup><https://www.live-resto.fr/>

<sup>2</sup><https://nbk-cg.com/>

<sup>3</sup>Une fintech néo-banque est une entreprise qui combine les services bancaires traditionnels avec la technologie et l'innovation pour fournir des solutions financières numériques.

<sup>4</sup><https://www.lislup.com/>

<sup>5</sup>[https://dz.linkedin.com/company/infoware-informatique?trk=public\\_profile\\_topcard-current-company](https://dz.linkedin.com/company/infoware-informatique?trk=public_profile_topcard-current-company)



### 1.3 Présentation du domaine métier

La Garde d'enfants ou babysitting, est une profession qui implique que l'exerçant prenne en charge les jeunes enfants à domicile (de bas âge jusqu'aux pré adolescents) en l'absence de leurs parents, pendant une courte ou longue durée [6].

Un gardien peut pratiquer plusieurs activités avec l'enfant. Cela peut varier selon son âge ou ses besoins. En effet, son travail ne se limite pas seulement à le nourrir ou à le surveiller pendant son sommeil, il peut aussi comprendre la préparation de ses repas, le soin de son hygiène personnelle ou même l'apport d'une bonne compagnie tout en assurant sa sécurité. Si on parle d'un enfant plus âgé, sa garde peut comprendre l'assistance aux devoirs, le transport depuis l'école ainsi que l'accompagnement aux activités extra scolaires (telles que les séances de sport ou de danse) [6].

Bien que ce terme soit lourdement féminisé, le rôle d'un gardien d'enfants peut aussi bien concerner les femmes que les hommes, le facteur déterminant reste avant tout l'expérience, les capacités à répondre aux exigences de la situation.

Il est à noter qu'en Algérie, une présentation de cette activité en fiche métier réglementaire officielle reste à définir.

#### 1.3.1 Les exigences du domaine

Exercer ce métier n'exige aucun diplôme ni agrément [5]. Cependant, un minimum d'expérience dans le milieu familial est requis étant donné qu'il s'agit d'un travail avec les petits.

Certaines capacités et compétences peuvent également être requises, telles que la patience, la communication, l'empathie ou certaines connaissances psychopédagogiques et sanitaires. L'obtention d'un diplôme ou d'un certificat de formation est donc avantageux au pratiquant de ce métier [6].

Les gardiens doivent aussi être responsables, fiables et bienveillants pour assurer la sécurité et le bien-être des enfants et ainsi tranquilliser les parents [6].

#### 1.3.2 Intérêts du domaine

Le domaine de la garde d'enfants présente plusieurs intérêts importants pour les parents et même pour les prestataires.

Pour les parents, cette solution :

- Apporte un soutien précieux en leur offrant des services flexibles qui s'adaptent facilement avec leurs plannings de travail ; les parents peuvent ainsi se concentrer sur leurs tâches ou même de profiter de leur temps libre en toute sérénité.
- Met à disposition un service pendant les moments critiques ou d'urgence.
- Permet à leurs enfants de bénéficier d'une garde personnalisée avec des activités adaptées, ce qui favorisera leur bon développement.

Pour les gardiens, cela génère de nombreuses opportunités d'emploi pour les personnes qui souhaitent travailler avec les enfants, ce qui contribue à stimuler l'économie locale.

### 1.3.3 Les types de garde d'enfants à domicile

On distingue différents modes de garde d'enfants à domicile, chacun propose une option pour répondre aux besoins spécifiques des parents, on citera [12] :

1. **Garde fréquente** : dédiée aux personnes qui s'occuperont des enfants à temps plein et donc pendant toute la semaine.
  - **Nourrice à temps plein** : elle prend l'enfant totalement en charge, elle peut l'assister dans des activités éducatives ou lui offrir des soins médicaux si nécessaire.
  - **Assistante maternelle** : également appelée auxiliaire maternelle, c'est une professionnelle de la petite enfance agréée par la direction de wilaya chargée de l'action sociale, qui accueille un nombre limité d'enfants à son domicile. Elle doit répondre aux conditions d'accueil qui lui sont imposées notamment en matière de sécurité et d'hygiène [7] .
2. **Garde occasionnelle** : soit quelques heures par jour ou certains jours de la semaine
  - **Nourrice à temps partiel** : elle ne travaille que pour quelques heures pendant la semaine, généralement moins de 40 heures.
  - **Babysitter** : elle est relativement moins expérimentée et peut être plus jeune qu'une nourrice ou une assistante maternelle, elle s'occupe de garder l'enfant occasionnellement pendant quelques heures par exemple durant le week-end ou le soir.
3. **Garde partagée** : à titre d'exemple, deux familles décident de partager les services d'une personne pour la garde de leurs enfants à domicile soit une nourrice ou bien une Babysitter. Cela permet de réduire les coûts et d'offrir une compagnie aux enfants.

Tous les types de garde d'enfants décrits précédemment peuvent être résumés dans la figure 1.2 explicative des différentes options qui s'offrent aux parents selon le mode de garde qu'ils choisissent, la principale différence entre ces modes réside dans la durée et la régularité des heures de travail.

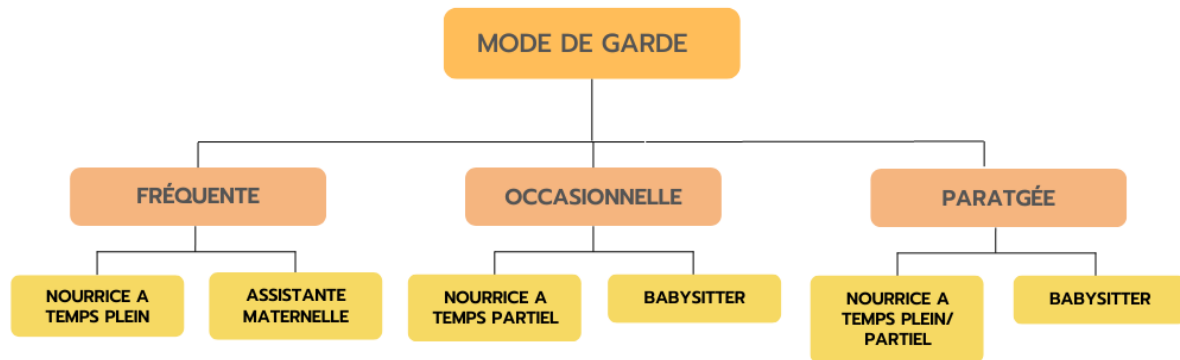


FIG. 1.2 : Types de garde d'enfants à domicile.

### 1.3.4 Recrutement

Pour les nourrices et les assistantes maternelles, un contrat est mis en place avec les parents, afin de clarifier les attentes et les responsabilités de chacun, ainsi que les modalités de paiement et les heures de travail.

Pour les babysitters, la prestation est journalière et donc ne nécessite aucun contrat avec le parent. Dans le but de mieux appréhender le paysage concurrentiel du domaine auquel se rapporte notre projet, après recherche nous relevons quelques sites tout en présentant une comparaison rassemblant leurs fonctionnalités essentielles.

## 1.4 Quelques sites de garde de référence

Nous avons procédé à une analyse approfondie afin de repérer et d'évaluer les concurrents pertinents, qu'ils soient nationaux ou internationaux.

### 1.4.1 Au niveau national

En Algérie, on trouve quelques offres d'emplois/services comme [Oudkniss.com](http://Oudkniss.com) ou des forums comme [Greataupair.com](http://Greataupair.com) ou [Loozap.com](http://Loozap.com) qui permettent aux prestataires d'offrir leurs services et aux parents de les rechercher.

On peut aussi trouver ce genre d'annonces sur les réseaux sociaux principalement sur [Facebook](https://www.facebook.com). Il n'existe cependant pas d'application qui se spécialise dans ce domaine.

### 1.4.2 Au niveau international

Plusieurs sites sont disponibles sur le marché, nous en citerons quelques-uns tels que [Babysits.com](http://Babysits.com), [NannyLane.com](http://NannyLane.com), [Sittercity.com](http://Sittercity.com), [Nounou.ma](http://Nounou.ma), [Nounou-top.fr](http://Nounou-top.fr), et [Care.com](http://Care.com). Ci-après le tableau 1.1 présentant quelques sites et leurs services à l'échelle internationale.

TAB. 1.1 : Tableau comparatif de quelques sites de gardes d'enfants.

Nom du site	Type de services	Application mobile	Services premium
Care.com	<ul style="list-style-type: none"> <li>• Garde d'enfants / bébés (inclut une recherche guidée)</li> <li>• Cours particuliers</li> <li>• Entretien de maison</li> <li>• Garde de personnes âgées</li> <li>• Garde d'animaux</li> </ul>	✓	✓
Sittercity.com	<ul style="list-style-type: none"> <li>• Garde d'enfants / bébés</li> <li>• Cours particuliers</li> <li>• Activités à l'extérieur</li> </ul>	✓	✓
NannyLane.com	<ul style="list-style-type: none"> <li>• Garde d'enfants / bébés à temps plein ou partiel</li> <li>• Offre aussi une recherche assistée</li> </ul>		✓
Nounou.ma	<ul style="list-style-type: none"> <li>• Garde d'enfants / bébés à temps plein ou partiel</li> </ul>		
Nounou-top.fr	<ul style="list-style-type: none"> <li>• Garde d'enfants / bébés</li> <li>• Cours particuliers</li> <li>• Entretien de maison</li> <li>• Garde de personnes âgées</li> <li>• Garde d'animaux</li> </ul>		

Les sites care.com, nannylane.com et Sittercity.com proposent des services premiums, Sittercity.com et care.com sont avantagés par rapport à nannylane.com car ils offrent une application mobile pour leur site, tandis que nounou.ma et nounou-top.fr disposent uniquement de sites web , un service commun à tous, celui de la garde d'enfant. Les fonctionnalités varient selon les sites, mais les plus basiques et récurrentes sont :

- La recherche de professionnels en fonction de la localisation, des disponibilités et des compétences.
- La consultation des profils des gardiens (incluant leur expérience, leurs certifications et les avis des autres parents).
- Un système de notation et la possibilité de laisser des commentaires.
- La gestion des contrats de garde.
- L'option de permettre aux prestataires de l'application d'utiliser les fonctionnalités de bases gratuitement.

Nous avons conduit une enquête auprès de quelques crèches de notre proximité (voir annexe D.1) et nous avons conclu que le processus le plus courant pour trouver un gardien pour son enfant est par le biais des recommandations faites de bouche à oreille, il et donc d'intérêt d'apporter une solution technologique à ce domaine au niveau national.

### 1.5 Problématique

Naturellement, parmi les services de garde d'enfants à disposition, les crèches traditionnelles sont la première solution à laquelle les parents ont recours, car elles représentent un environnement propice à l'épanouissement de leurs enfants avec des individus de même âge et des personnes qualifiées pour les guider tout en assurant leur sécurité. Cependant, d'après les résultats de nos recherches, nous avons remarqué un déséquilibre entre l'offre de la part des services de garde d'enfants et la demande des parents. Selon ces derniers : "Les conditions d'accueil ne sont pas toujours convenables"[2], voici quelques raisons poussant les parents à tenir ces propos :

- Les crèches d'enfants offrent des horaires fixes, qui varient généralement de 7h-7h30 jusqu'à 16h30-17h30. Cela peut paraître peu pratique pour certains parents dont les horaires de travail ne concordent pas tout à fait avec ceux de ces dernières. Ajoutons à cela que, certains établissements ordonnent un tarif supplémentaire (allant de 200 à 500 DA l'heure) en cas de retard [8].
- Le nombre de crèches en Algérie n'est pas suffisant par rapport au nombre d'enfants qui nécessitent d'y séjourner. A titre d'exemple, dans la wilaya de Béjaia chef lieu, on compte une vingtaine d'établissements pour une population d'environ un million d'habitants [13] [11].

- Les crèches n'acceptent qu'un nombre très limité d'enfants, il arrive souvent que les parents fassent le parcours du combattant pour trouver une place pour leurs enfants, dans la majorité des cas ils prennent l'initiative de le faire avant le début des vacances d'été [1].
- Les coûts des services s'avèrent être élevés, essentiellement dans les crèches privées, qui offrent néanmoins des services de meilleure qualité [3]. « Les crèches imposent des tarifs élevés, pour ne pas dire exorbitants. La fourchette par enfant se situe entre 8 000 DA et 12 000 DA. En effet, les frais d'inscription, l'assurance et les frais des manuels sont autant de charges imposées aux parents dès le départ. Et il faut galérer pour trouver une bonne crèche » [3].
- Les crèches doivent être soumises à certaines normes, comme celles du "décret 02-87 de 17 septembre 2008 conformément au cahier des charges relatif aux normes pédagogiques d'espace, volume d'air, hygiène et sanitaires" [2], Cependant, il se trouve qu'une bonne partie de ces établissements ne les respecte pas [2], et cela affecte négativement la santé des enfants, en favorisant la manifestation de différentes maladies telles que les indigestions, les poux, etc.

### 1.6 Solution envisagée

Dans la problématique soulevée précédemment, nous avons mentionné les difficultés que les parents rencontrent vis à vis des services de garde d'enfants au niveau des crèches. Il s'avère qu'une solution alternative existante s'offre à eux, notamment les service de garde d'enfants à domicile.

Notre projet vise à concevoir et mettre en œuvre une plate-forme <sup>6</sup> nommée "Kodomo", qui sera un environnement numérique où les parents pourront facilement, rapidement et en toute sécurité demander des services de garde adaptés aux besoins spécifiques de leurs enfants. Elle offrira également aux gardiens l'opportunité de trouver des emplois dans ce domaine.

### 1.7 Conclusion

Dans ce chapitre, nous avons présenté l'organisme d'accueil ainsi que l'étude de l'existant du domaine métier. Nous avons ainsi constaté que la garde d'enfants est une préoccupation majeure pour de nombreuses familles. Par ailleurs certains défis et lacunes, relevés à travers nos recherches, confrontent ces familles dans leurs quêtes de gardiens fiables et compétents. Notre objectif est de contribuer à la création d'une plate-forme optimisée qui simplifiera la recherche et la demande de services de garde de haute qualité. Dans les chapitres suivants, nous approfondirons notre compréhension du sujet et présenterons notre méthodologie de développement afin de concevoir une solution répondant aux attentes des parents et aux exigences du marché.

---

<sup>6</sup>Environnement permettant la gestion ou l'utilisation de services applicatifs

## Chapitre 2

# Langages de modélisation et méthodologies de développement

### 2.1 Introduction

Dans ce chapitre, nous examinerons en détail l'utilisation du langage de modélisation UML pour représenter et visualiser notre solution. Nous aborderons également les méthodologies de développement Scrum et le processus unifié (UP) que nous avons choisi de fusionner et d'intégrer dans notre approche de conception. À cet effet il sera mis en évidence l'impact de ces outils pour apporter clarté, précision et efficacité, qui sont conditions de bonne conduite de notre travail ; ainsi que la communication et la collaboration au sein de notre équipe de développement. Nous expliquerons en quoi leur utilisation a été pertinente pour notre projet.

### 2.2 Langages de modélisation

Un langage de modélisation est un langage spécifique utilisé pour représenter graphiquement ou textuellement différents aspects d'un système ou d'un processus. Ces langages permettent de communiquer, de documenter et de comprendre les différents éléments et relations d'un système. Dans le cadre de ce projet nous nous intéresserons qu'à UML et OCL.

#### 2.2.1 Définition d'UML

Le langage de modélisation informatique UML (Unified Modeling Language) est une référence pour la modélisation objet et la programmation orientée objet. Il permet de représenter graphiquement des éléments du monde réel ou virtuel sous la forme d'objets informatiques [4] (p.1).

UML utilise des diagrammes pour décrire la structure et le comportement des objets dans un système, offrant une présentation simplifiée et facilement compréhensible des systèmes logiciels complexes. UML est utilisé non seulement dans le développement de logiciels, mais également dans d'autres domaines tels que l'industrie, l'ingénierie et le marketing pour la modélisation de processus [20].

#### 2.2.2 Définition d'OCL

OCL (Object Constraint Language) est un langage de contraintes utilisé dans le cadre du langage de modélisation UML, fournissant une syntaxe formelle pour exprimer des expressions logiques, des prédicats, des opérations et des contraintes sur les classes, les attributs, les opérations, les associations et donc sur les modèles UML ; dans le but de clarifier les règles métier, de vérifier la conformité de ces modèles et faciliter la génération de code à partir de ces derniers.



### 2.2.3 Diagrammes utilisés

UML définit 14 types de diagrammes divisés en deux catégories : structure et comportement. Cependant, nous allons nous limiter qu'à quelques uns d'entre eux : [4] (p.10-12).

**Diagrammes de structure :** Diagramme de classes.

**Diagrammes de comportement :** Diagramme de cas d'utilisations, diagramme de séquence, diagramme d'activité.

## 2.3 Méthodologie de développement

Une méthodologie de développement génie logiciel est un ensemble de principes, de pratiques et de processus qui guident le développement de logiciels de manière efficace et structurée. Ces méthodologies fournissent le cadre et les méthodes de gestion de projets, de planification, de conception, de mise en œuvre, de test et de maintenance des logiciels [14](p.44).

Nous avons opté pour l'usage d'une méthodologie hybride qui regroupe SCRUM et Unified Process (UP). Bien que ces deux méthodes soient différentes, elles peuvent être combinées pour améliorer l'efficacité et la qualité du processus de développement logiciel.

### 2.3.1 SCRUM

Scrum est un cadre de travail (Framework en anglais) de gestion de projets vastes, il se centre sur un but commun et permet de répondre de manière productive et créative à des problèmes complexes par son évolutivité, en sus livrant des produits de qualité. Il sert donc à livrer le produit de manière incrémentale et itérative en incluant le client durant tout le processus de développement, ce qui apporte une amélioration continue au produit pour finalement atteindre la satisfaction du client [16].

#### Création du BackLog

Un Backlog (ou carnet de produit), est la liste de toutes les fonctionnalités, les améliorations et les corrections à apporter à un produit ou un projet, en se concentrant sur les besoins utilisateurs que l'on décrira en forme de "User Stories" <sup>1</sup>. Cette liste va être organisée par ordre de priorité afin de pouvoir débiter par les fonctionnalités les plus importantes [15] (p.33).

Il est important de mentionner que le Backlog est un document dynamique qui évolue au cours du temps, à mesure que le produit est développé et que de nouvelles informations sont acquises [15] (p.33).

---

<sup>1</sup>sous forme de courtes descriptions simples, histoire utilisateur en français.

### L'équipe SCRUM

L'équipe Scrum est un groupe de personnes qui travaillent ensemble pour développer un produit ou un projet à l'aide de la méthode Scrum. Elle est composée de trois rôles principaux : le Product Owner, le Scrum Master et l'équipe de développement [16].

- **Product Owner** : Responsable de produit, est la personne qui définit le Backlog, elle s'assure que le produit final répond bien aux besoins des utilisateurs [15] (p.34).
- **SCRUM Master** : Responsable d'organisation, il s'assure que l'équipe respecte bien les règles de SCRUM, il l'aide à résoudre certains problèmes et à améliorer sa productivité [15] (p.32).
- **Equipe de développement** : Se charge d'exécuter les tâches du développement.

### Découpage du projet selon SCRUM

Un sprint équivaut à un cycle de développement dans le cadre de Scrum. Il s'agit d'une itération qui se déroule sur une période de temps fixe, généralement de **deux à quatre** semaines, pendant laquelle l'équipe de développement travaille sur un ensemble limité de tâches spécifiques et prédéfinies [9].

1. Le sprint commence par une réunion de planification, où l'équipe définit les objectifs et sélectionne les éléments du carnet de produit (backlog) à développer pour l'itération en cours.
2. Pendant le sprint, l'équipe de Scrum travaille sur les tâches qui ont été identifiées et se réunit quotidiennement pour des "stand-up meetings", des réunions permettant de suivre l'avancement du travail et de résoudre rapidement les éventuels problèmes qui peuvent survenir.
3. À la fin de chaque sprint, l'équipe de développement participe à une étape de révision où ses membres présentent les résultats de leur travail aux parties prenantes, et celles-ci peuvent donner lieu à un feedback.

Ci-dessous 2.1, un schéma explicatif du fonctionnement de SCRUM.

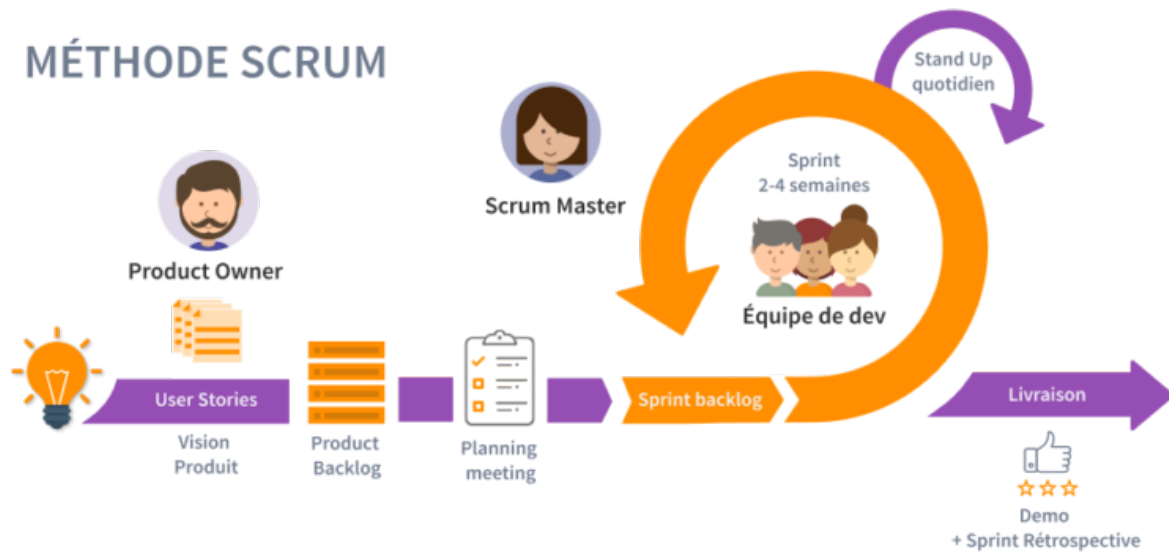


FIG. 2.1 : Schéma explicatif du fonctionnement de Scrum [10].

### 2.3.2 Processus unifié (UP)

Le Processus Unifié (PU ou UP, pour Unified Process en anglais) est une méthode de développement logiciel itérative et incrémentale qui se compose de quatre phases principales : **la création, l'élaboration, la construction et la transition** [18].

Chaque phase a des objectifs spécifiques, des activités et des livrables. Ces phases sont répétées plusieurs fois jusqu'à ce que le système soit développé et jugé satisfaisant [18].

#### Quelques principes de UP

On retient trois principes clés du processus unifié : [17]

- L'approche centrée sur l'architecture du logiciel.
- il est piloté par les cas d'utilisation.
- il Comprend le développement itératif et incrémental.

### 2.3.3 Combinaison des deux méthodes

La combinaison de UP (Unified Process) et Scrum est une approche qui permet d'exploiter les avantages des deux méthodologies pour obtenir un processus de développement adapté aux besoins de la plate-forme. Voir tableau récapitulatif 2.1.

TAB. 2.1 : Avantages des deux méthodologies.

Method	Avantages
Scrum	<ul style="list-style-type: none"><li>- Les itérations courtes des Sprints.</li><li>- Gestion du Backlog.</li><li>- Création de l'équipe Scrum.</li><li>- La livraison régulière.</li><li>- Interactions entre les membres d'équipe.</li><li>- facilité d'adaptation aux nouvelles exigences.</li></ul>
Processus Unifié	<ul style="list-style-type: none"><li>- Itérations structurées avec des jalons définis pour chaque phase.</li><li>- La Planification détaillée.</li><li>- La gestion des risques ( identification, analyse, mesures à prendre).</li><li>- La documentation détaillée.</li></ul>

UP fournit une structure de processus pour la planification, la conception, la modélisation et la documentation. Combinée avec Scrum, notre équipe de développement peut élaborer un plan détaillé pour chaque étape du projet, tout en utilisant des sprints pour livrer des fonctionnalités opérationnelles de manière régulière et rapide [15] [17].

## 2.4 Conclusion

L'utilisation conjointe des langages UML et OCL, ainsi que l'application des méthodologies de développement Scrum et UP, vont jouer un rôle essentiel dans notre approche de développement. Ils nous permettront de mieux appréhender les besoins et les attentes des utilisateurs, de réaliser une modélisation proche de la réalité.

Les prochains chapitres détailleront les différentes étapes de notre processus de développement, en mettant en avant les pratiques spécifiques des méthodologies utilisées pour parvenir à une issue réussie.

# Chapitre 3

## Sprint zéro

### 3.1 Introduction

Dans ce chapitre, on passe en revue, l'ensemble du déroulement du sprint 0. C'est la phase introductive de Scrum, qui peut être équivalente à la phase de "création" dans UP, elle permet de préparer le terrain pour le projet à venir. Cela inclut la planification initiale, la sélection de l'équipe, ainsi que la planification du backlog du produit.

Dans cette étape également, l'équipe de développement met en place l'infrastructure, l'environnement de développement et les outils nécessaires pour le projet. Cela inclut la mise en place des serveurs, l'installation des logiciels et la configuration des outils de gestion de projet. Le sprint 0 est aussi l'occasion de définir les approches de développement et la gestion des risques.

### 3.2 Expression des besoins

C'est une étape clé du processus de développement logiciel. Elle vise à comprendre et à documenter en détail les exigences et les attentes des parties prenantes pour le système logiciel à développer [4] (p.29). et pour ce faire, on procédera comme suit :

#### 3.2.1 Identification des acteurs

Les acteurs d'un système sont des entités externes au système qui interagissent avec lui pour réaliser un objectif spécifique. Les acteurs peuvent être des personnes, des organisations, des systèmes informatiques, des dispositifs matériels [4] (p.29-30).

C'est en identifiant les acteurs de notre système et en collaborant avec les parties prenantes, qu'on pourra recueillir les besoins fonctionnels, non fonctionnels, ainsi que les cas d'utilisations [4] (p.99-102).

Le système que nous avons nommé "KODOMO" comprendra les acteurs suivants :

1. **Administrateur**
2. **Gestionnaire**
3. **Parent**
4. **Gardien**

#### 3.2.2 Diagramme de contexte

Bien qu'il ne soit pas reconnu en tant que diagramme d'UML, il est conventionnel en ingénierie logicielle de commencer la conception par un diagramme de contexte. Ce dernier fournit une vue globale des interactions entre le système et ses acteurs [19]. Voir figure 3.1.

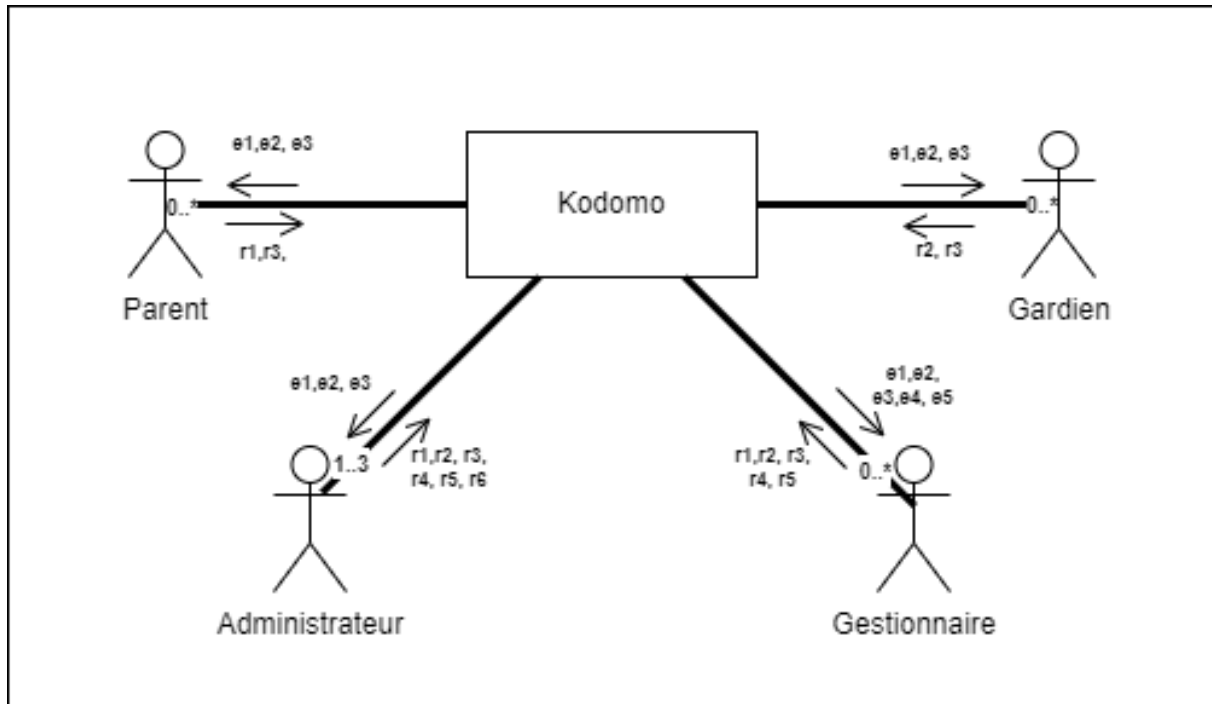


FIG. 3.1 : Diagramme de contexte dynamique.

### Description des messages échangés

Le diagramme de contexte précédent 3.1 illustre les quatre acteurs qui utilisent le système. Pour chaque acteur, des flux d'échanges avec celui-ci sont observés. Les flèches sortantes du système représentent les messages émis aux acteurs, tandis que les flèches entrantes représentent les messages reçus par le système. Le tableau 3.1 récapitule l'ensemble de ces échanges.

TAB. 3.1 : Tableau de messages échangés.

ID	Description
<b>Messages émis par le système</b>	
e1	Afficher toutes les données du système (Parents, gardien, candidatures, offres de garde, gestionnaires..)
e2	Informé le bon enregistrement d'une donnée.
e3	Afficher un message d'erreur système.
<b>Messages reçus par le système</b>	
r1	Demande de création d'offre de garde
r2	Demande de candidature
r3	Recherche et filtrage de données auxquelles l'utilisateur veut accéder

Table 3.1 Tableau de messages échangés. (suite)

ID	Description
r4	Demande d'ajout, modification et suppression des parents, gardiens, candidatures, offres de garde, etc.
r5	Validation d'un entretien
r6	Demande d'ajout d'un gestionnaire

### 3.2.3 Équipe Scrum

L'équipe Scrum implique le Product Owner, le Scrum Master, et une équipe de développement.

Dans le cadre de notre stage, le Product Owner est un personne de l'entreprise d'accueil : Monsieur *Reda Bekka*. Il est responsable de la définition du produit à développer. Notre équipe de développement travaille en étroite collaboration avec lui pour s'assurer que les fonctionnalités développées répondent aux besoins des utilisateurs et des parties prenantes. Quant à la position de SCRUM Master, elle a été remplie par l'un des chefs de projet de l'entreprise, Mme *Samia Kherbouche* et notre encadrant M.*Khaled Allem* . Enfin, l'équipe de développement nous a inclut en tant que stagiaires dans l'entreprise, nommément *Esma Hocini* et *Meriem Yasmine Touati*.

Schématiquement, l'organisation selon les principes de SCRUM se résument ainsi :

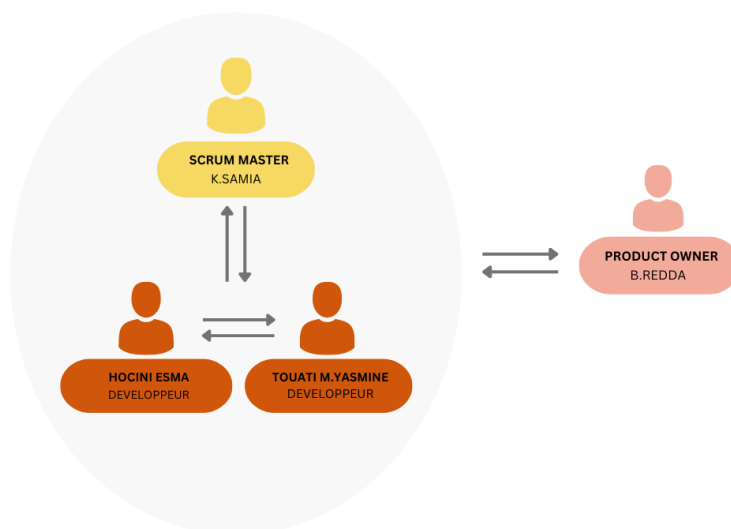


FIG. 3.2 : Constitution de l'équipe Scrum.



### 3.2.4 Définition des besoins fonctionnels

Dans cette section, nous allons capturer les besoins fonctionnels de la plate-forme afin de mieux concevoir la solution, nous commencerons par la spécification des cas d'utilisation.

#### Les cas d'utilisation

Les cas d'utilisation décrivent les différentes fonctionnalités et les actions spécifiques que les utilisateurs peuvent effectuer avec le système. Ces cas d'utilisation seront ensuite représentés par un diagramme de cas d'utilisation montrant l'ensemble des acteurs, les interactions et les étapes du scénario de chaque cas [4] (p.99-102).

NOTE Notre plate-forme utilise l'infrastructure super administrateur - administrateur (respectivement administrateur - gestionnaire pour notre cas). Les administrateurs ont des privilèges étendus et gèrent globalement le système, tandis que les gestionnaires ont des responsabilités spécifiques pour la gestion des utilisateurs et des contenus. Cela permet une répartition des tâches et un contrôle précis des autorisations. On peut subdiviser les fonctions de chaque acteur comme suit :

#### 1. Administrateur, ce dernier a pour tâches de :

- Gérer les comptes clients (parents ou gardiens), inclut toutes les opérations d'ajout, de lecture, de suppression et de modification.
- Bloquer le compte d'un client (parents ou gardiens)
- Gérer les entretiens de garde entre un parent et un gardien
- Créer une tâche de garde.
- Traiter le profil d'un nouveau gardien
- Surveiller l'état du système
- Gérer les gestionnaires

#### 2. Gestionnaire aux tâches suivantes :

- Gérer le compte d'un client (parents ou gardiens).
- Bloquer un client (parents ou gardiens).
- Gérer un entretien de garde entre un parent et un gardien.
- Créer une tâche de garde.
- Traiter le profil d'un nouveau gardien.
- Consulter les statistique du système.

#### 3. Parent aux fonctionnalités suivantes :

- Créer une offre de garde.
- Signaler un gardien.

- Évaluer un gardien.
- Initier une demande d'entretien.

#### 4. Gardien :

- Postuler pour une offre d'emploi.

### Diagramme de cas d'utilisation global

On illustre les fonctionnalités précédentes dans le diagramme de cas d'utilisation 3.3 suivant.

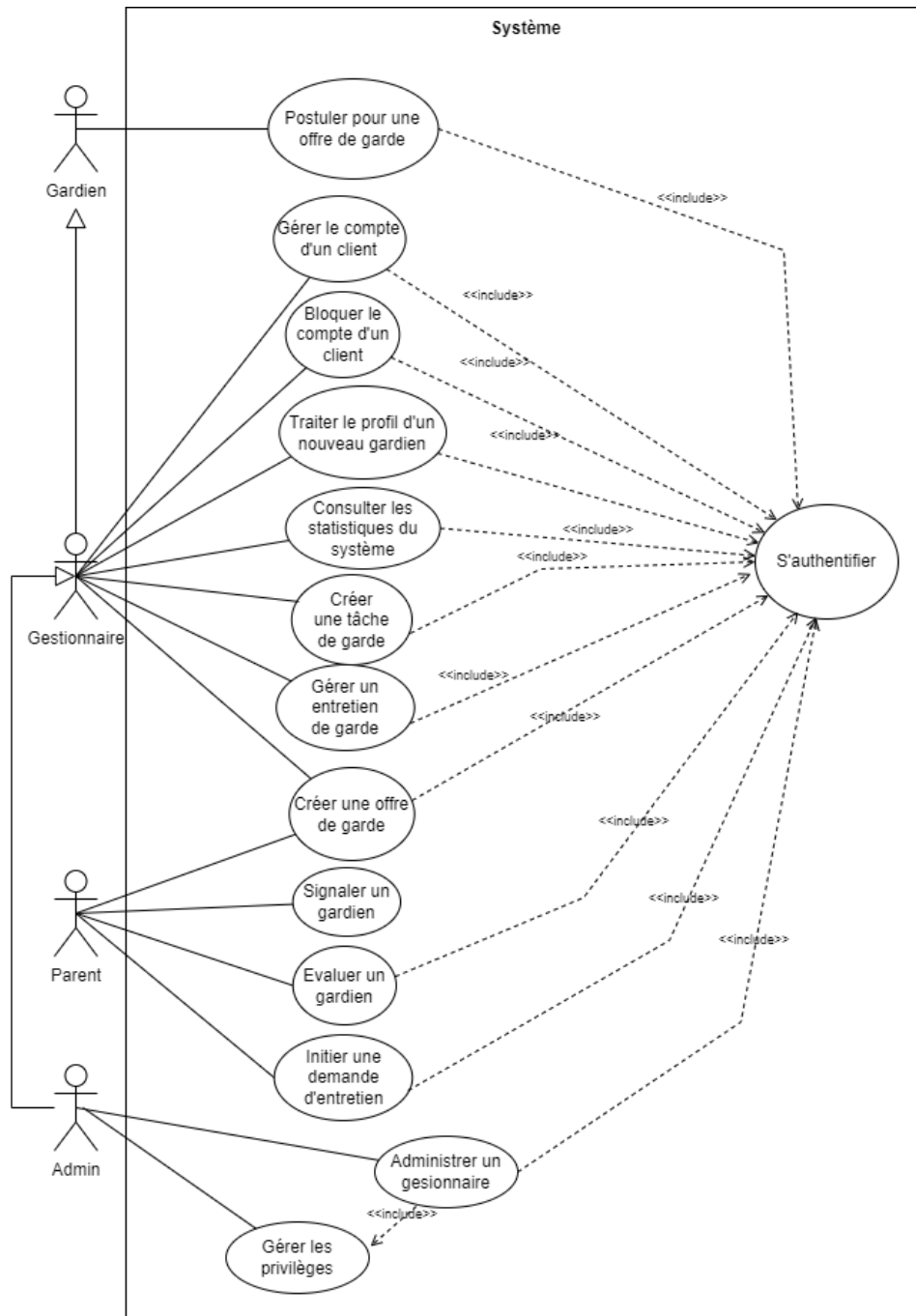


FIG. 3.3 : Diagramme de cas d'utilisation global.

### Spécification non fonctionnelles

Ce genre d'exigence se concentre sur la manière dont le système doit fonctionner et sur les attributs de qualité qui doivent être satisfaits, à savoir :

1. Les interfaces doivent s'adapter pour une utilisation sur toutes les tailles d'écran (responsives).
2. Ajouter le mode sombre, pour améliorer la lisibilité de l'utilisateur dans des conditions de faible luminosité.

3. Design moderne et attrayant pour une expérience utilisateur agréable.
4. La possibilité de basculer vers d'autres langues ( Avec l'anglais et le français disponibles comme options actuellement).
5. Mises à jour régulières pour améliorer les fonctionnalités et la sécurité du site, en restant d'actualité avec les tendances technologiques et en se référant aux retours des utilisateurs à des itérations conséquentes.
6. Les informations personnelles des utilisateurs sont des données sensibles, elles doivent être stockées, cryptées et protégées contre les accès non autorisés.

### 3.3 Planification avec Scrum

Nous avons mentionné dans le chapitre précédent, que l'organisation de notre processus de développement suivait les principes de Scrum. Dans cette section, nous expliquerons davantage comment nous avons appliqué cette méthodologie pour la gestion de nos tâches au sein de notre équipe.

#### 3.3.1 Création du Backlog de produit

Comme évoqué précédemment dans la section 2.3.1 du chapitre 3, la responsabilité de pourvoir le Backlog revient au Product Owner. Après avoir eu des discussions avec lui, nous avons pu recueillir les attentes et les besoins des utilisateurs sous forme de User Stories, chacune est associée à une priorité, ainsi qu'à un niveau de risque. Le tableau 3.2 suivant présente un Product Backlog global.

TAB. 3.2 : Le Backlog du produit général.

ID	User Story	Priorité	Risque
US-01	"En tant d'utilisateur, je souhaite posséder un compte, et y accéder en m'authentifiant."	Élevée	Moyen
US-02	"En tant que administrateur, je souhaite pouvoir gérer les comptes des parents et des gardiens."	Élevée	Faible
US-03	"En tant que administrateur, je veux pouvoir vérifier et traiter le profil d'un nouveau gardien avant qu'il puisse proposer ses services auprès des parents."	Élevée	Important
US-04	"En tant que administrateur, je peux pénaliser un client (parent ou gardien) en bloquant son compte."	Élevée	Faible
US-05	"En tant que administrateur je souhaite pouvoir consulter les statistiques de l'application."	Basse	Faible
US-06	"En tant que administrateur, je peux ajouter un gestionnaire et gérer ses privilèges vis à vis des différentes fonctionnalités du système."	Élevée	Moyenne
US-07	"En tant que administrateur, je peux créer la liste des tâches que les gardiens peuvent effectuer avec les enfants."	Élevée	Faible
US-08	"En tant que administrateur, je pourrais organiser un rendez-vous pour un entretien physique entre le parent et une nourrice à une date et heure de mon choix."	Élevée	Important
US-09	"En tant que parent, je souhaite pouvoir créer des offres de garde."	Élevée	Moyen
US-11	"En tant que parent, je souhaite initier une demande d'entretien avec un gardien choisi."	Élevée	Moyen
US-12	"En tant que gardien, je souhaite pouvoir trouver et postuler à une offre de garde."	Élevée	Moyen

### 3.3.2 Planification des sprints

Dans le tableau 3.3 ci-dessous, on illustre la répartition chronologique des sprints. Au total, nous en avons effectué trois (3), chacun ayant une durée de 2 à 4 semaines. Dans le premier Sprint nous nous sommes focalisées sur les fonctionnalités de gestion des parents et des gardiens, le deuxième Sprint introduit le gestionnaire en tant que nouvel acteur du système avec la gestion de ses privilèges, et enfin, dans le troisième Sprint nous avons implémenté les fonctionnalités des gestions des offres de garde, des postulations ainsi que les rendez-vous d'entretiens entre les parents et les gardiens.

Numéro du sprint	Date début	Date fin
SPRINT 1	20/03/2023	15/04/2023
SPRINT 2	17/04/2023	03/05/2023
SPRINT 3	06/05/2023	07/06/2023

TAB. 3.3 : Tableau des dates du début et fin de chaque sprint

### 3.4 Présentation du logo de la plate-forme

Voici le logo 3.4 que nous avons mis en oeuvre pour la plate-forme "KODOMO". À noter que KODOMO signifie enfant en langue japonaise.



FIG. 3.4 : Logo de KODOMO

### 3.5 Configuration de l'environnement de développement

Plusieurs outils sont mis à notre disposition pour faciliter le processus de réalisation d'une application informatique, afin de bien mener ce projet jusqu'à son aboutissement nous avons utilisé les outils suivants :

### 3.5.1 Bibliothèques utilisées

Nous avons utilisé des bibliothèques JavaScript pour notre projet, React.js côté client et Nest.js côté serveur.

#### ReactJS

ReactJS <sup>1</sup>est une bibliothèque JavaScript open-source créée par Facebook, que nous avons utilisée pour développer le FrontEnd de notre système, elle utilise une syntaxe appelée JSX, qui permet d'écrire du code HTML dans du code JavaScript, permettant de créer des composants réutilisables qui peuvent être assemblés pour construire des interfaces utilisateur complexes. Sa facilité de maintenance, sa rapidité d'exécution, et son efficacité en terme de mémoire, en font une des bibliothèques les plus populaires de nos jours.

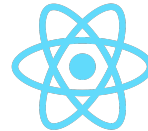


FIG. 3.5 : Logo de ReactJS.

#### NestJS

NestJS <sup>2</sup> est un framework Node.js. Nous l'avons choisi car elle propose une grande variété de fonctionnalités et de modules pour développer la partie BackEnd, tels que la gestion des requêtes et des réponses HTTP, la validation des données, l'authentification, la sécurité, la gestion des fichiers, les websockets, la planification des tâches, les bases de données, et bien plus.



FIG. 3.6 : Logo de NestJS.

### 3.5.2 Environnement de développement

VSC <sup>3</sup>pour Visual Studio Code est un éditeur de code source. Il est gratuit et open-source, et offre des fonctionnalités telles que la coloration syntaxique, la complétion de code, la gestion de versions, le débogage, les tâches personnalisées et l'intégration avec des outils externes tels que Git, Type Script et Node.js. VSCode est également extensible grâce à un large écosystème de plugins, qui permettent aux développeurs d'ajouter de nouvelles fonctionnalités et de personnaliser l'éditeur en fonction de leurs besoins.



FIG. 3.7 : Logo de VSC.

---

<sup>1</sup><https://react.dev/learn> (visité le 10/05/2023).

<sup>2</sup><https://docs.nestjs.com/> (visité le 10/05/2023).

<sup>3</sup><https://code.visualstudio.com/> (visité le 10/05/2023).

### Outil de modélisation

**Draw.io**<sup>4</sup> est un logiciel de cartographie mentale en ligne qui permet aux utilisateurs de créer différents types de graphiques, tels que des diagrammes, des organigrammes, des flux de processus, etc. Il est utilisé pour organiser visuellement les idées et les concepts sur le fonctionnement d'un projet, facilitant ainsi la communication et la compréhension. Nous allons faire usage de cet outil pour modéliser les diagrammes d'UML ainsi que pour les différents schémas explicatifs dans le mémoire.

### 3.5.3 Quelques outils d'organisation et de planification

Pour nous aider à mieux communiquer et à nous organiser, nous avons fait usage de quelques outils tels que :

- Telegram<sup>5</sup>, une application de messagerie instantanée pour la communication en toute sécurité avec l'ensemble de l'équipe Scrum.
- GitHub<sup>6</sup> une plate-forme de développement de logiciels pour stocker et gérer notre projet en ligne afin de mieux collaborer avec d'autres développeurs de l'équipe.
- ClickUp<sup>7</sup> une plate-forme de gestion de projet et de collaboration en ligne pour le découpage, la spécification et le suivi des tâches par les membres de l'entreprise.

### 3.5.4 Outils de test et de documentation

Afin de valider le bon fonctionnement de l'API, nous avons testé les différents End-points<sup>8</sup> en utilisant Postman, nous avons aussi généré leurs documentation détaillée à l'aide de l'outil Swagger.

- Postman<sup>9</sup> est un logiciel de collaboration pour le développement d'API. Il permet aux développeurs de tester et de déboguer des API, de créer des requêtes et des collections d'API, et de partager facilement leur travail avec leurs équipes.
- Swagger<sup>10</sup> est une librairie open source qui permet de concevoir, de documenter et de tester des API REST de manière simple et efficace. Il permet de créer une documentation interactive pour les API, ce qui facilite la compréhension de leur fonctionnement par les utilisateurs.

---

<sup>4</sup><https://www.drawio.com/> (visité le 15/06/2023).

<sup>5</sup><https://telegram.org/> (visité le 10/05/2023).

<sup>6</sup><https://github.com/> (visité le 10/05/2023).

<sup>7</sup><https://clickup.com/> (visité le 10/05/2023).

<sup>8</sup>également appelés points de terminaison, sont des URL spécifiques d'une API auxquelles les clients peuvent envoyer des requêtes pour interagir avec le système ou récupérer des données.

<sup>9</sup><https://www.postman.com/> (visité le 10/05/2023).

<sup>10</sup><https://swagger.io/> (visité le 10/05/2023).



### 3.6 Conclusion

L'objectif de l'étape d'expression des besoins est de créer une base solide pour le développement du système en comprenant les attentes des parties prenantes, en définissant les fonctionnalités essentielles et en établissant les bases pour la conception et les implémentations ultérieures. Cette étape est cruciale pour s'assurer que le logiciel répondra aux besoins et aux objectifs du projet. Dans le prochain chapitre, nous entamerons notre travail en commençant par le premier sprint.

# Chapitre 4

## Sprint un

## 4.1 Introduction

Le présent chapitre se concentre sur le sprint 1 de notre projet. Au cours de cette itération, nous allons exposer la mise en œuvre d'un ensemble de fonctionnalités clés, en nous basant sur les exigences et les priorités définies dans le Backlog du produit (tab. 3.2) lors du sprint zéro. Dans cette section, nous décrirons en détail les objectifs, les activités et les réalisations de ce sprint, en nous basant sur les disciplines de UP, c'est à dire les étapes de spécification et d'analyse, de conception, puis d'implémentation et des tests.

## 4.2 Backlog du Sprint 1

Les User Stories définies dans le Product Backlog, fournissent un moyen simple et facile à comprendre pour que l'équipe de développement puisse identifier les fonctionnalités sur lesquelles travailler. A chaque itération, on sélectionne une liste des fonctionnalités sur lesquelles on compte travailler. Le tableau suivant montre le Backlog du sprint 1, et représente la liste des tâches que nous avons sélectionné à partir du Product Backlog, et qui doivent être terminés pendant le sprint.

Lors du Sprint 1, nous nous sommes focalisées sur les fonctionnalités qui concernent la session d'authentification, rafraîchissement de la session, changement du mot de passe, ainsi que la gestion des parents et des gardiens par l'administrateur. Ces fonctionnalités sont illustrées dans le tableau 4.2 qui suit :

TAB. 4.1 : Backlog du Sprint 1.

ID User Story	ID Tâche	Tâches	Durée
US-1	1.1	Élaborer les CU et description de la fonctionnalité "S'authentifier". Ajouter les diagrammes de séquence et ajouter les classes de conception nécessaires.	5h
	1.2	Développer et tester les fonctionnalités.	32h
	2.1	Élaborer le CU de la fonctionnalité "Gérer le compte d'un gardien". Ajouter le diagramme de séquence et les classes de conception nécessaires.	7h
	2.2	Développer et tester la fonctionnalité.	40h

TAB. 4.2 : Backlog du Sprint 1. (suite)

US-2	2.3	Élaborer le CU de la fonctionnalité "Gérer le compte d'un parent". Ajouter le diagramme de séquence et les classes de conception nécessaires.	6h
	2.4	Développer et tester la fonctionnalité.	35h
US-3	3.1	Élaborer le CU de la fonctionnalité "Traiter le profil d'un nouveau gardien". Ajouter le diagramme de séquence et les classes de conception nécessaires.	4h
	3.2	Développer et tester la fonctionnalité.	10h
US-4	4.1	Élaborer le CU et la description de la fonctionnalité "Bloquer un gardien". Ajouter les diagrammes de séquence et les classes de conception nécessaires.	3h
	4.2	Développer et tester la fonctionnalité.	10h
	4.3	Élaborer le CU et la description de la fonctionnalité "Bloquer un parent". Ajouter les diagrammes de séquence et les classes de conception nécessaires.	3h
	4.4	Développer et tester la fonctionnalité.	10h

### 4.3 Spécifications

La phase de spécification est fondamentale car elle nous permet de définir les objectifs, les fonctionnalités et les contraintes du système à développer. Une fois les spécifications validées, elles serviront de guide pour les phases de conception, de développement et de test ultérieures. Dans les spécifications sus citées nous nous intéressons aux points suivants :

### 4.3.1 Diagrammes de cas d'utilisation

Le diagramme 4.1 suivant englobe la modélisation des cas d'utilisation déduits à partir du premier Sprint :

#### Sprint 1

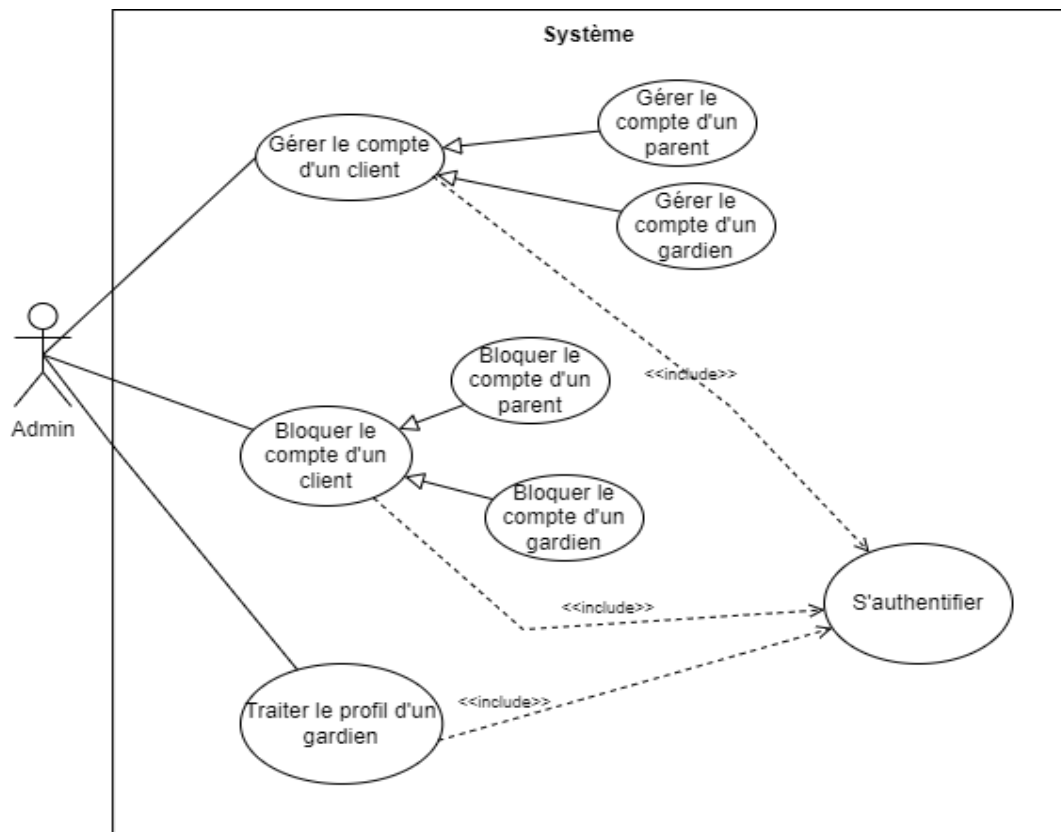


FIG. 4.1 : Diagramme de cas d'utilisation du Sprint 1

### 4.3.2 Descriptions textuelles

Dans cette partie, nous allons introduire la documentation des cas d'utilisation les plus pertinents du Sprint 1, ainsi que quelques règles afférentes aux cas d'utilisation.

### Cas d'utilisation "S'authentifier"

TAB. 4.3 : Règles de gestion : S'authentifier.

ID	Règles de gestion
RG-001	L'email doit exister et doit être valide.
RG-002	Le mot de passe doit être identique à celui de la base de données.

TAB. 4.4 : Description textuelle du CU : S'authentifier.

<b>CU : S'authentifier</b>
<b>ID : 1</b>
<b>Description brève :</b> L'étape primaire pour accéder au système.
<b>Acteurs primaires :</b> Tous
<b>Pré condition :</b> L'utilisateur doit posséder un compte
<b>Enchaînement principal :</b> Le cas d'utilisation démarre lorsque l'utilisateur accède au système. <ol style="list-style-type: none"><li>1. L'utilisateur saisit son e-mail et mot de passe.</li><li>2. L'utilisateur clique sur « se connecter » et attend la fin du chargement de l'opération.</li><li>3. Le système valide l'authentification de l'utilisateur.</li></ol>
<b>Post condition :</b> L'utilisateur obtient accès à sa session.
<b>Enchaînement alternatif :</b> <ol style="list-style-type: none"><li>3.1. Si le mot de passe ne correspond pas à celui de sa session, l'application indique à l'utilisateur que les informations d'identification sont incorrectes et invite l'utilisateur à saisir de nouveau ses informations d'identification. Le scénario reprend au point 1.</li><li>3.2.1. L'utilisateur oublie son mot de passe et clique sur le bouton "<b>Mot de passe oublié</b>".</li><li>3.2.2. Le système affiche un petit formulaire lui demandant de saisir l'e-mail associé à son compte.</li><li>3.2.3 L'utilisateur effectue la saisie, et envoie les informations.</li><li>3.2.4 Le système envoie un message vers la boîte e-mail de l'utilisateur, contenant un lien.</li><li>3.2.5 L'utilisateur, à la réception du message, clique sur le lien qui le ré-dirigera vers la page de l'application, contenant un formulaire de saisie du nouveau mot de passe avec la confirmation de ce dernier.</li></ol>

3.2.6 L'utilisateur effectue la saisie, et la valide.

3.2.7 Le système enregistre les informations.

Le scénario reprend au point 1.

3.3. L'utilisateur se connecte avec un e-mail inexistant, le système affiche une erreur indiquant que l'e-mail n'existe pas dans la base de données.

### Cas d'utilisation "Gérer le compte d'un client"

TAB. 4.5 : Description textuelle du CU : Gérer le compte d'un client.

Gérer le compte d'un client
<b>ID</b> : 2
<b>Description brève</b> : Permettre au gestionnaire du système de gérer les informations et les actions relatives à un parent ou un gardien dans l'application.
<b>Acteurs primaires</b> : Gestionnaire.
<b>Acteurs secondaires</b> : Parent, gardien
<b>Pré condition</b> : <ol style="list-style-type: none"><li>1. Le gestionnaire doit être connecté.</li><li>2. Le client à gérer doit exister dans la base de données.</li></ol>
<b>Enchaînement principal</b> : <ol style="list-style-type: none"><li>1. Le gestionnaire choisit quel type de client à gérer (parent ou gardien).<ol style="list-style-type: none"><li>A. Le gestionnaire souhaite créer un nouveau compte client.</li></ol></li><li>2. Le système affiche le formulaire des informations nécessaires à remplir.</li><li>3. Le gestionnaire saisit et valide les informations.</li><li>4. Le système enregistre le nouveau compte et affiche un message de succès.<ol style="list-style-type: none"><li>B. Le gestionnaire souhaite modifier le compte d'un client existant.<ol style="list-style-type: none"><li>2. Le gestionnaire peut rechercher le client spécifique qu'il souhaite gérer, en utilisant des critères de recherche tels que le nom, l'adresse e-mail.</li><li>3. Le système affiche une liste des clients correspondants aux critères de recherche.</li><li>4. Le gestionnaire sélectionne le client souhaité dans la liste.</li><li>5. Le système affiche les informations détaillées du client, telles que son nom, ses coordonnées, etc.</li></ol></li></ol></li></ol>

TAB. 4.6 : Description textuelle du CU : Gérer le compte d'un client. (suite)

<p>6. Le gestionnaire peut effectuer les actions suivantes :</p> <ol style="list-style-type: none"><li>Sélectionner et modifier ses informations, telles que son nom, son adresse, son numéro de téléphone, etc.</li><li>Supprimer son compte .</li></ol> <p>7. Une fois les modifications effectuées, l'administrateur peut enregistrer les changements dans le système.</p> <p>Le système met à jour les informations du parent avec les modifications apportées.</p>
<p><b>Post condition</b> : Les modifications apportées sont mises à jour dans le système.</p>
<p><b>Enchaînement alternatif</b> :</p> <ol style="list-style-type: none"><li>3.1. Le gestionnaire ajoute un client avec une adresse e-mail existante dans la base de données, le système affiche un message l'informant que le client existe déjà.</li><li>3.2. Lors de la modification, le gestionnaire saisit des informations invalides, le système affiche un message l'informant que certaines informations sont invalides.</li><li>3.3. Le gestionnaire souhaite supprimer un client dont le statut est "validé(e)", le système affiche un message l'informant que ce n'est pas possible de supprimer un utilisateur actif.</li></ol>

### Cas d'utilisation "Traiter le profil d'un gardien"

TAB. 4.7 : Description textuelle du CU : Traiter le profil d'un gardien.

Traiter le profil d'un gardien
<b>ID</b> : 3
<b>Description brève</b> : Le gestionnaire peut vérifier le profil d'un gardien pour décider de l'éligibilité de son profil.
<b>Acteurs primaires</b> : Gestionnaire
<b>Acteurs secondaires</b> :
<b>Pré condition</b> : Le gestionnaire doit être connecté.
<p><b>Enchaînement principal</b> :</p> <p>Le cas d'utilisation commence lorsque le gestionnaire est notifié qu'un gardien vient de s'inscrire dans l'application.</p> <ol style="list-style-type: none"><li>1. Le gestionnaire accède au profil du nouveau gardien.</li><li>2. L'application affiche toutes les informations le concernant.</li></ol>



TAB. 4.8 : Description textuelle du CU : Traiter le profil d'un gardien. (suite)

3. Le gestionnaire effectue les vérifications par rapport aux informations personnelles et professionnelles du gardien.
4. Le gestionnaire valide le profil.
5. Le gestionnaire met le statut du profil du gardien à "actif(ve)".
<b>Post condition</b> : Le gardien est notifié de la décision du gestionnaire par e-mail et pourra utiliser son compte.
<b>Enchaînement alternatif</b> :
4.1.1. Le gestionnaire ne valide pas le profil.
4.1.2. L'application met le statut du profil du gardien à "refusé(e)".

### Cas d'utilisation "Bloquer le compte d'un client"

TAB. 4.9 : Règles de gestion : Bannir un gardien.

ID	Règles de gestion
RG-003	L'administrateur ne peut pas bannir un gardien si il l'est déjà.
RG-004	L'administrateur ne peut pas bannir un gardien si il est refusé, ou en attente de validation.
RG-005	Le motif du bannissement ne doit pas être vide

TAB. 4.10 : Description textuelle du CU : Bloquer le compte d'un client.

<b>CU</b> : Bloquer le compte d'un client
<b>ID</b> : 4
<b>Description brève</b> : L'administrateur est sur le point de bloquer le compte d'un client
<b>Acteurs primaires</b> : Gestionnaire
<b>Acteurs secondaires</b> : Gardien, parent
<b>Pré condition</b> :
1. Le bannissement d'un client devrait être une décision prise après une enquête et une évaluation appropriées des comportements ou de signalements , conformément aux politiques et procédures de l'application.

TAB. 4.11 : Description textuelle du CU : Bloquer le compte d'un client. (suite)

<ol style="list-style-type: none"><li>2. Le gestionnaire doit être connecté.</li><li>3. Le client doit exister dans la base de données.</li></ol>
<p><b>Enchaînement principal :</b></p> <p>Le cas d'utilisation démarre lorsque le gestionnaire reçoit un signalement à propos du client.</p> <ol style="list-style-type: none"><li>1. Le gestionnaire choisit la section du client concerné (parent ou gardien).</li><li>2. Le gestionnaire recherche et sélectionne le compte du client en question.</li><li>3. Le système affiche le profil du client.</li><li>4. Le gestionnaire sélectionne l'option "Bloquer".</li><li>5. L'application affiche une fenêtre de confirmation demandant à l'administrateur de préciser le motif et de confirmer sa décision.</li><li>6. Le gestionnaire saisit l'information puis confirme la décision de bannir le client en sélectionnant "Confirmer".</li><li>7. L'application désactive le compte du client et l'empêche d'accéder à celle-ci et à ses fonctionnalités.</li></ol>
<p><b>Post condition :</b> Le client reçoit un e-mail le notifiant de son bannissement ainsi que du motif de cette mesure.</p>
<p><b>Enchaînement alternatif :</b></p> <ol style="list-style-type: none"><li>4.1. Si le gestionnaire souhaite débloquent un client, il clique sur le bouton "Débloquer".</li></ol>

## 4.4 Analyse

Après avoir relevé les besoins, leur analyse nous permet d'identifier les acteurs du système, les cas d'utilisation, les scénarios et les flux de travail. Ces éléments servent de base pour la conception et la modélisation ultérieures du système, afin de clarifier comment le système doit interagir avec son environnement. Dans ce qui suit nous présenterons le modèle du domaine du sprint 1 ainsi que quelques diagrammes de séquence système des cas d'utilisation les plus pertinents pour ce sprint.

### 4.4.1 Modèle de domaine

Un modèle de domaine aide à capturer la compréhension commune du domaine entre les parties prenantes. Il souligne les parties les plus pertinentes du métier tel que les comportements, relations entre entités et les attributs qui les définissent. Dans ce qui suit nous allons présenter les concepts clés du domaine rapportés à notre projet.

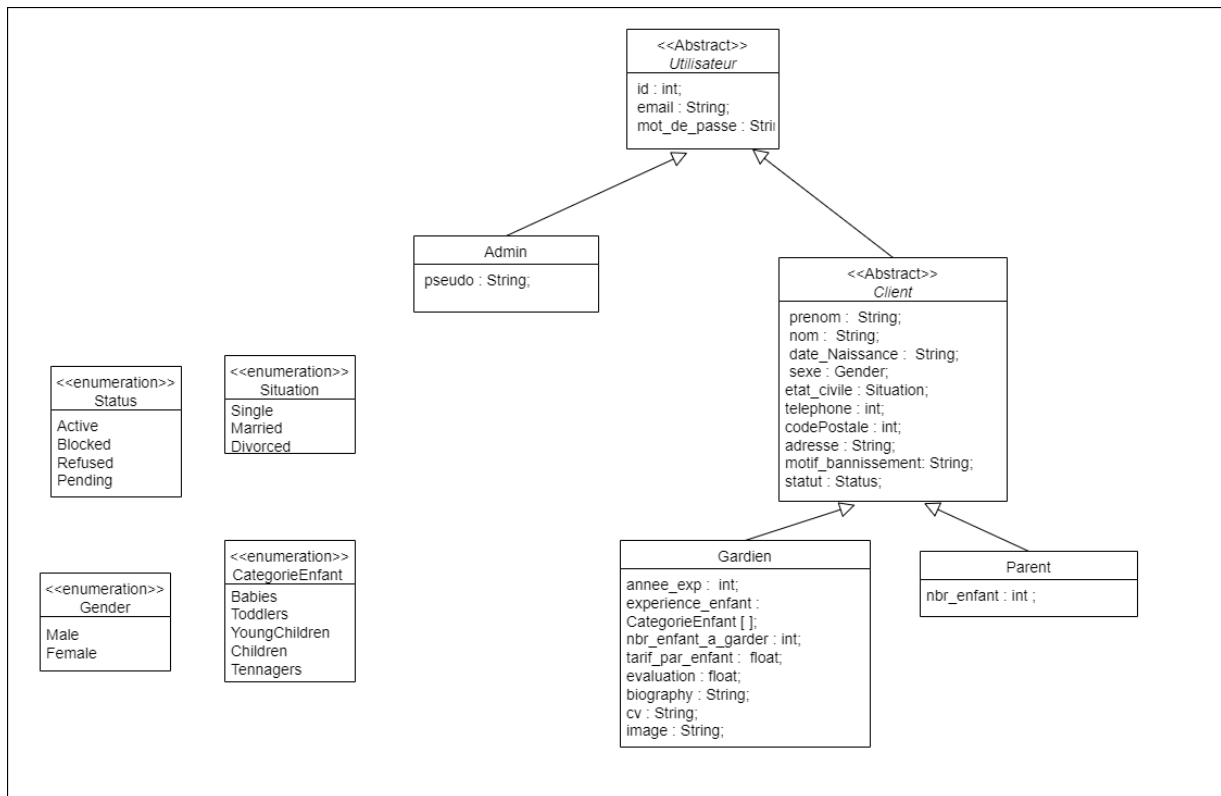


FIG. 4.2 : Modèle de domaine du sprint 1.

#### 4.4.2 Diagrammes de séquence système

Un diagramme de séquence système est utilisé pour représenter la séquence des messages échangés entre différents objets ou composants d'un système dans le cadre d'un scénario donné. Dans ce qui suit nous nous intéresserons aux cas les plus pertinents du sprint un.

##### Diagramme de séquence système d'authentification

Bien que trivial, le cas d'utilisation «s'authentifier» est indispensable à tout système informatique; l'utilisateur introduit ses identifiants, le système vérifie leur validité puis recherche ces derniers et les compare, et selon le résultat de la comparaison, l'utilisateur est soit authentifié soit réinvité à introduire de nouveau ses identifiants, le cas d'utilisation inclut d'autres cas selon certaines conditions.

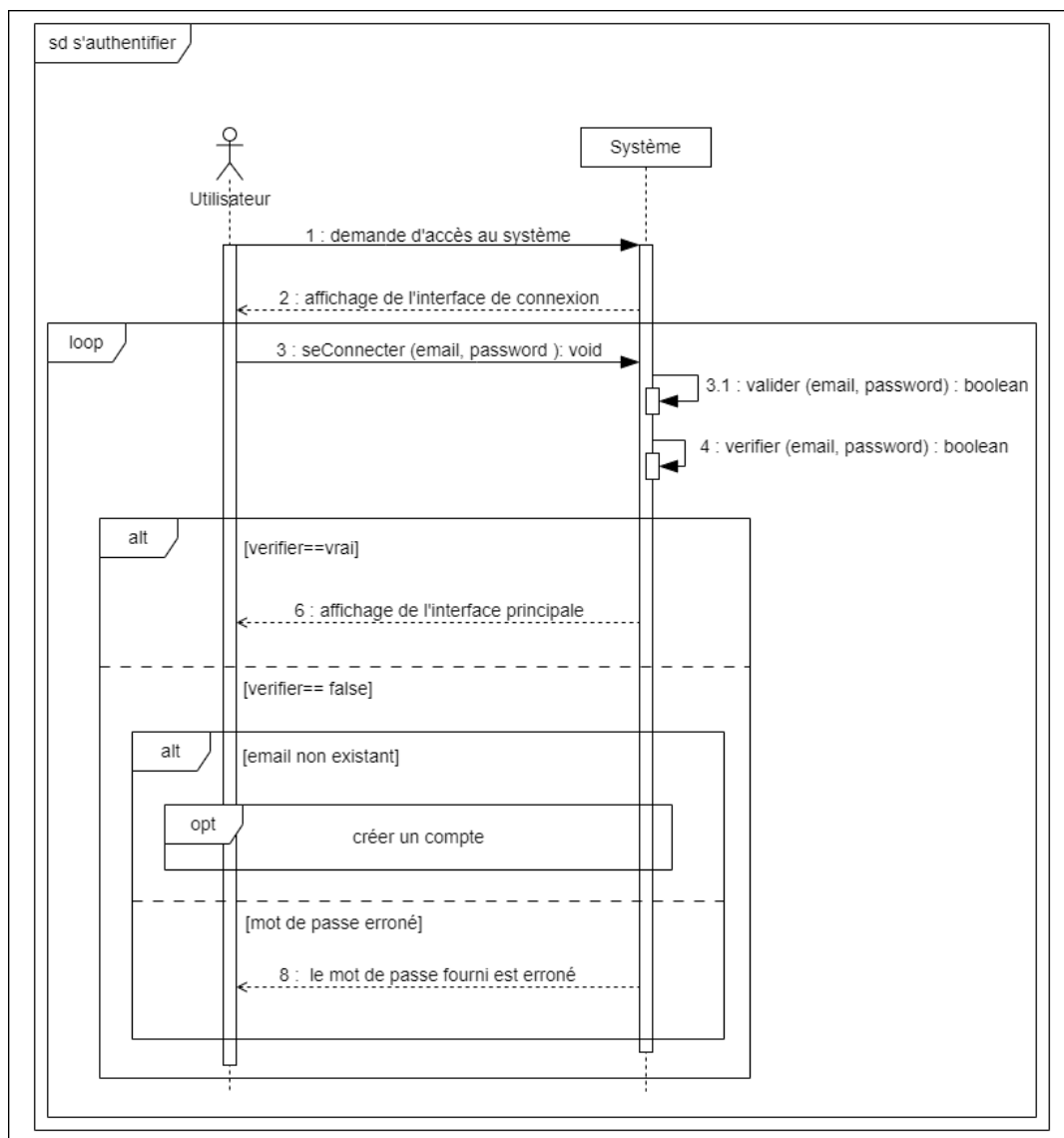


FIG. 4.3 : Diagramme de séquence système du cas d'utilisation «s'authentifier».

### Diagramme de séquence système d'administration d'un gestionnaire»

Le cas d'utilisation «administrer un gestionnaire» a pour tâche de créer un gestionnaire, définir ses privilèges ou bien de le supprimer. Lors de l'ajout la validation se fait en premier lieu puis une vérification sur l'existence d'un éventuel gestionnaire ayant un même email que celui introduit suivra.

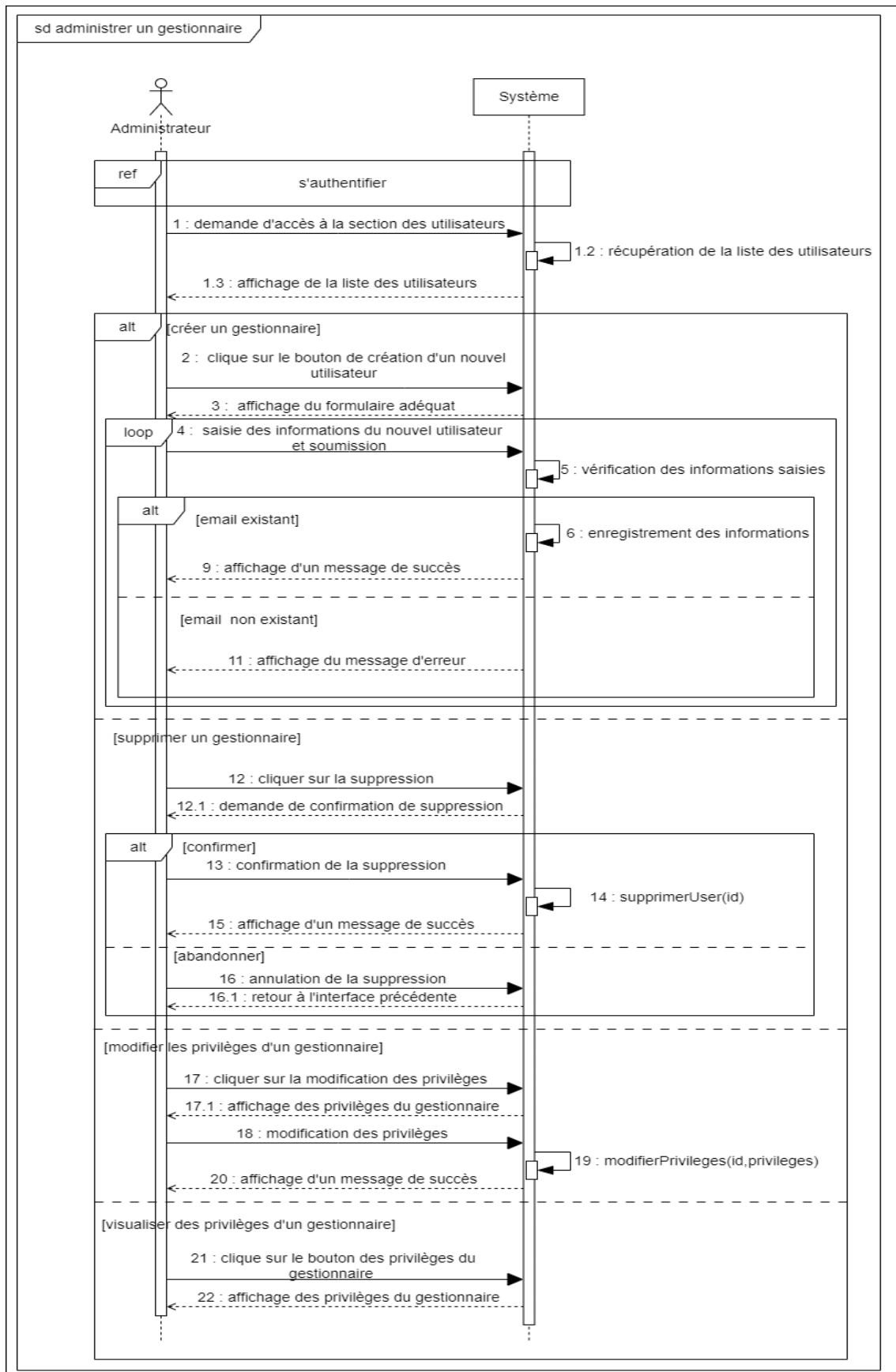


FIG. 4.4 : Diagramme de séquence système du cas d'utilisation «administrer un gestionnaire».

Diagramme de séquence système de traitement d'un profil de gardien

Le cas d'utilisation «traiter le profil d'un gardien» a pour but d'analyser et d'étudier le profil d'un gardien avant de valider sa postulation à l'application Kodomo.

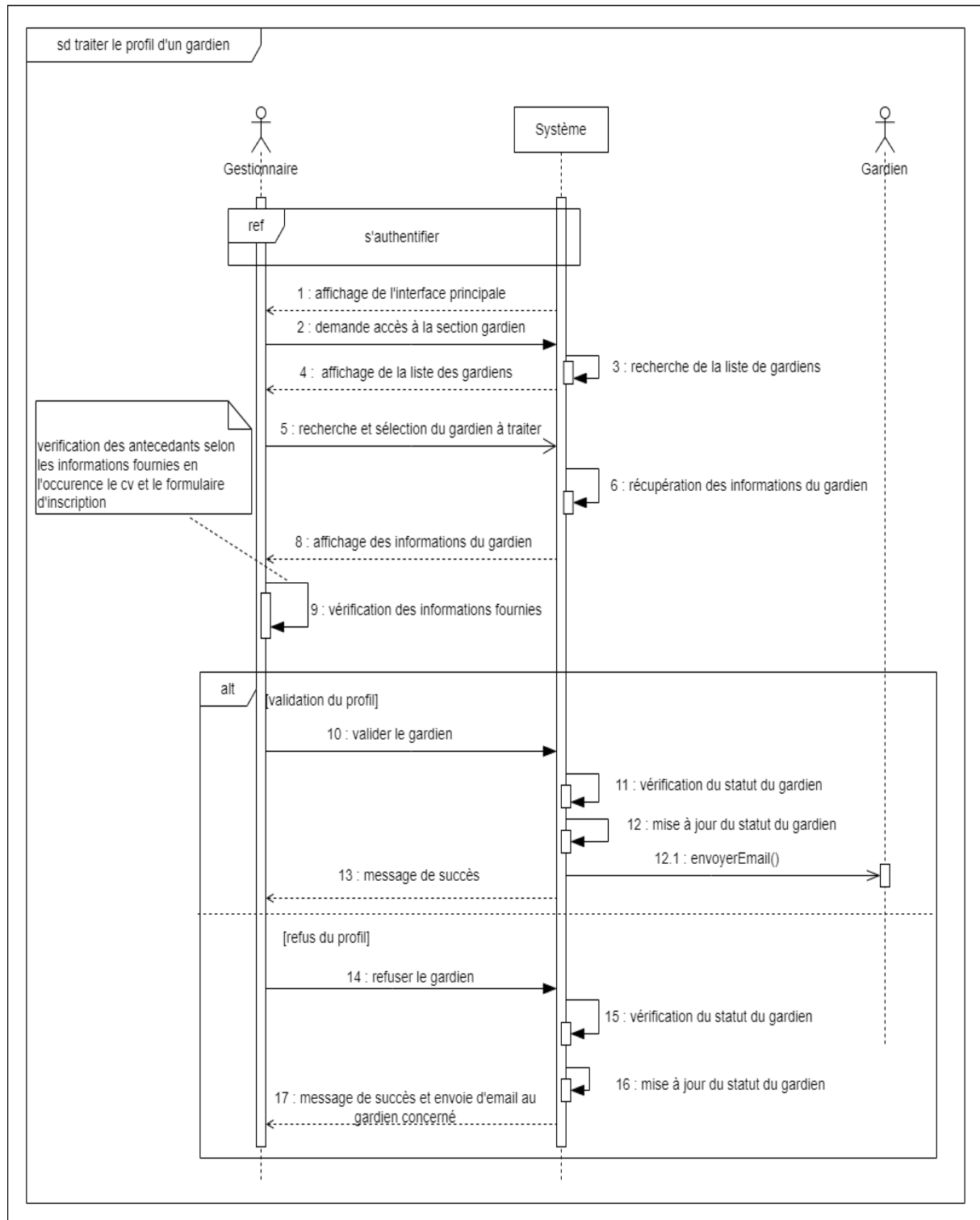


FIG. 4.5 : Diagramme de séquence système du cas d'utilisation «traiter le profil d'un gardien».

## 4.5 Conception

L'objectif de cette phase est de fournir une base solide pour le logiciel, en décrivant l'architecture, les composants et les détails de mise en œuvre nécessaires. Une bonne conception facilite la maintenance, l'extension et la compréhension du système, tout en garantissant la satisfaction des exigences spécifiées lors de la phase précédente.

### 4.5.1 Diagrammes de séquences détaillés

Un diagramme de séquences détaillés est utilisé pour représenter la séquence d'interactions survenant entre les objets eux-mêmes et/ou les acteurs eux-mêmes dans un système logiciel. Il met l'accent sur la chronologie et l'ordre des messages échangés entre les différents éléments du système, permettant ainsi de comprendre le flux de contrôle et les interactions entre les différents composants. Schématiquement pour illustrer ces situations nous nous sommes aidés de MVC (Modèle-Vue-Contrôleur), et nous avons noté : M : Modèle, UI : interface utilisateur, C : Contrôleur, comme suit :

#### Diagramme de séquence détaillé d'authentification

En plus du diagramme de séquence système, nous mettrons en relief le fonctionnement de l'authentification sous différents composants.

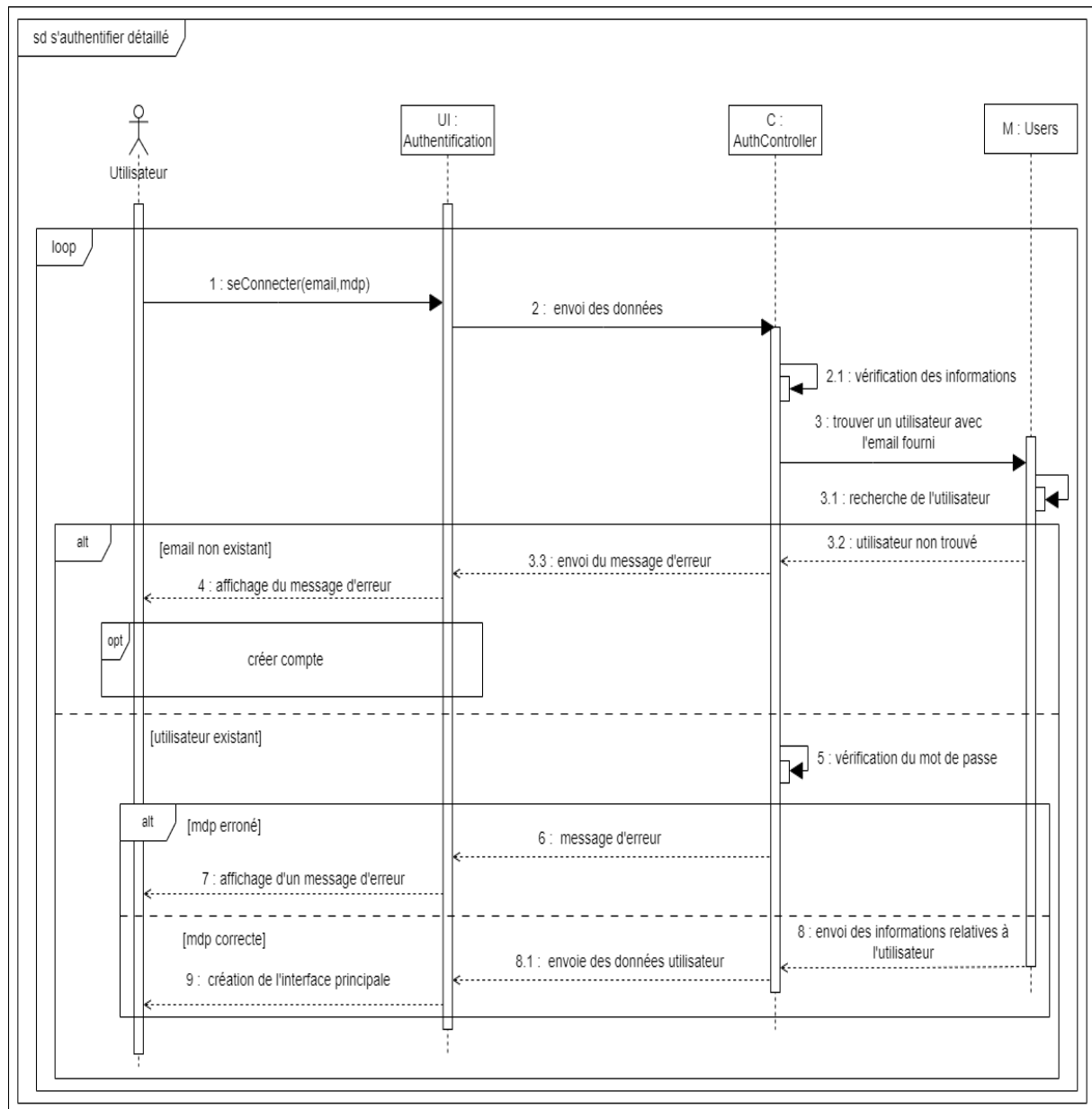


FIG. 4.6 : Diagramme de séquence système du cas d'utilisation «s’authentifier».

### 4.5.2 Diagrammes d’activité

Un diagramme d’activité est utilisé pour modéliser le comportement dynamique d’un système, en mettant l’accent sur les actions qui se déroulent au fil du temps. Il représente donc le flux d’activités, d’actions et de décisions dans un processus ou un scénario donné. Les diagrammes suivants sont la transcription du fonctionnement et du flux de chaque cas d’utilisation.



Diagramme d'activité d'authentification

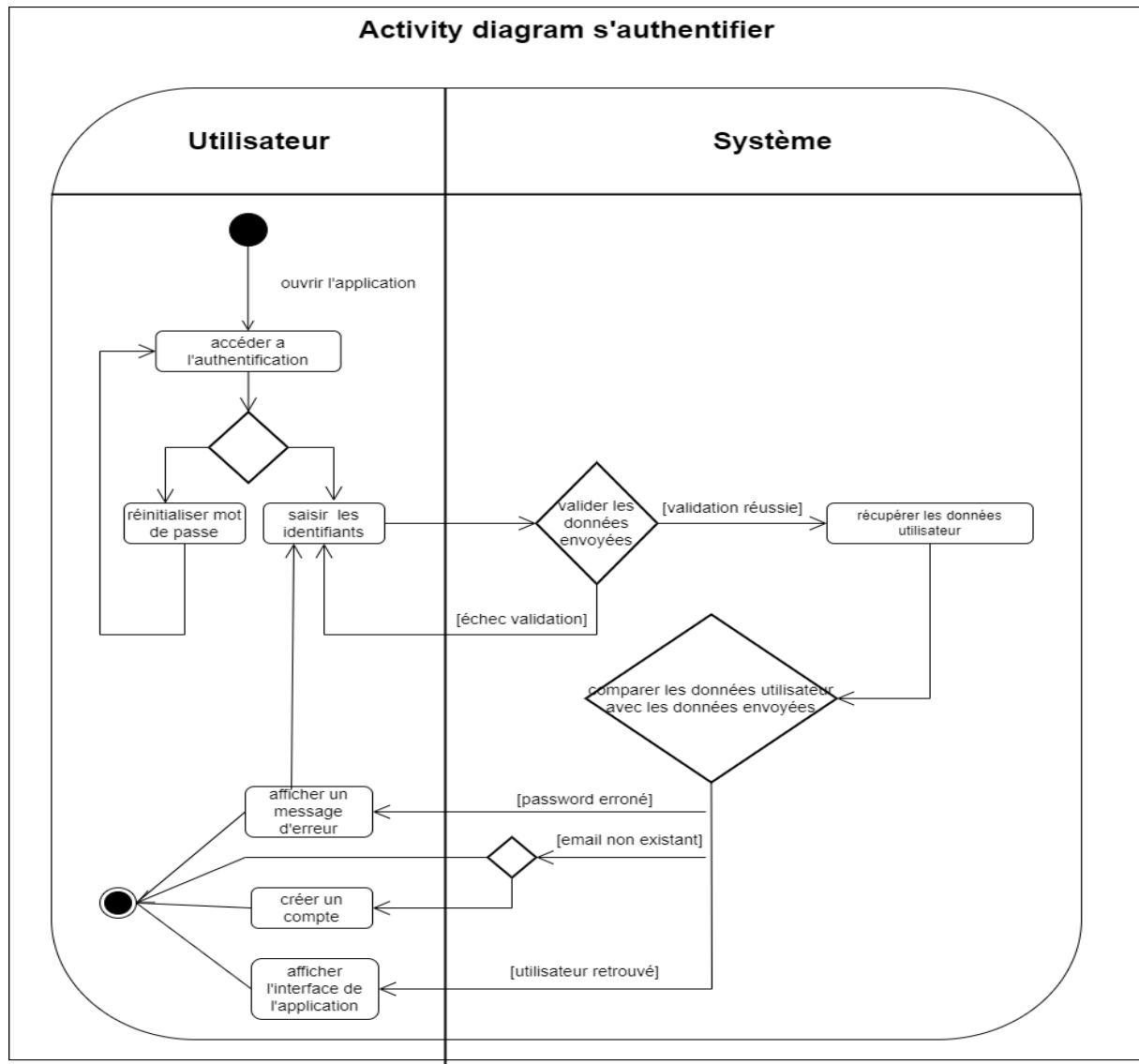


FIG. 4.7 : Diagramme d'activité du cas d'utilisation «s'authentifier».

### 4.5.3 Diagramme des classes de conception

La figure 4.8 représente le diagramme de classe conception du premier sprint où on peut voir les acteurs intervenants pendant ce sprint.

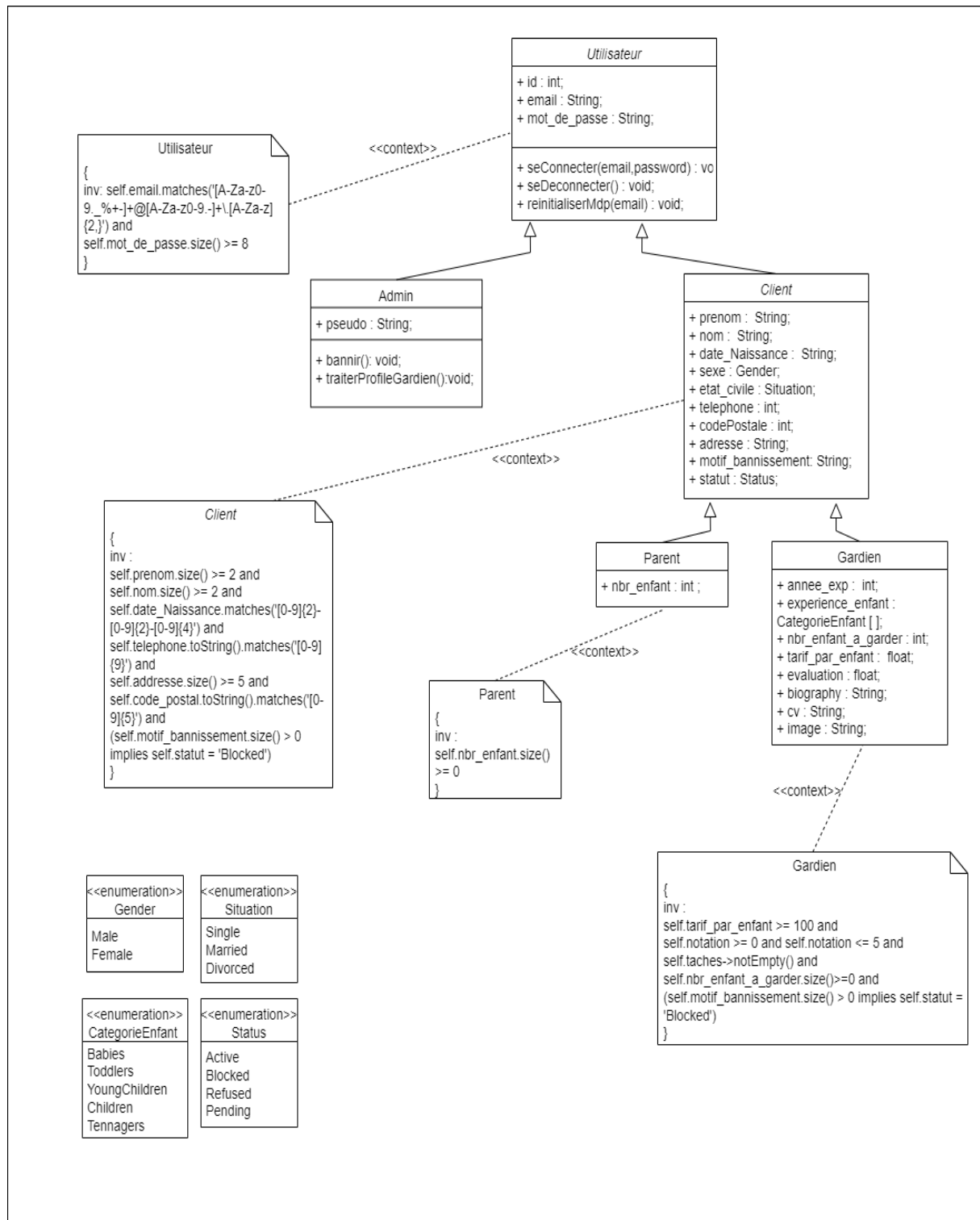


FIG. 4.8 : Diagramme de classes de conception du sprint 1.

## 4.6 Implémentation

Dans la phase d'implémentation, nous allons nous consacrer à la programmation en utilisant les langages ainsi que les bibliothèques appropriés, en suivant les spécifications et la conception détaillée pour mettre en œuvre les fonctionnalités et les comportements du

système. Nous soulignons que pour satisfaire le besoin non fonctionnel de sécurité, nous avons utilisé la librairie `bcrypt` basée sur la fonction de hashage `blowfish`, pour crypter et sécuriser les données sensibles.

### 4.6.1 Modèle relationnel de données

Par usage du `push-down` sur les classes abstraites `Utilisateur` et `Client`, nous obtenons le modèle relationnel suivant :

`Admin(id, pseudo, email, password, roles, privileges)`.

`Parent(id, email, nom, prenom, password, sexe, etat-civile, telephone, codePostale, adresse, nbr-enfant, motif-bannissement, statut)`.

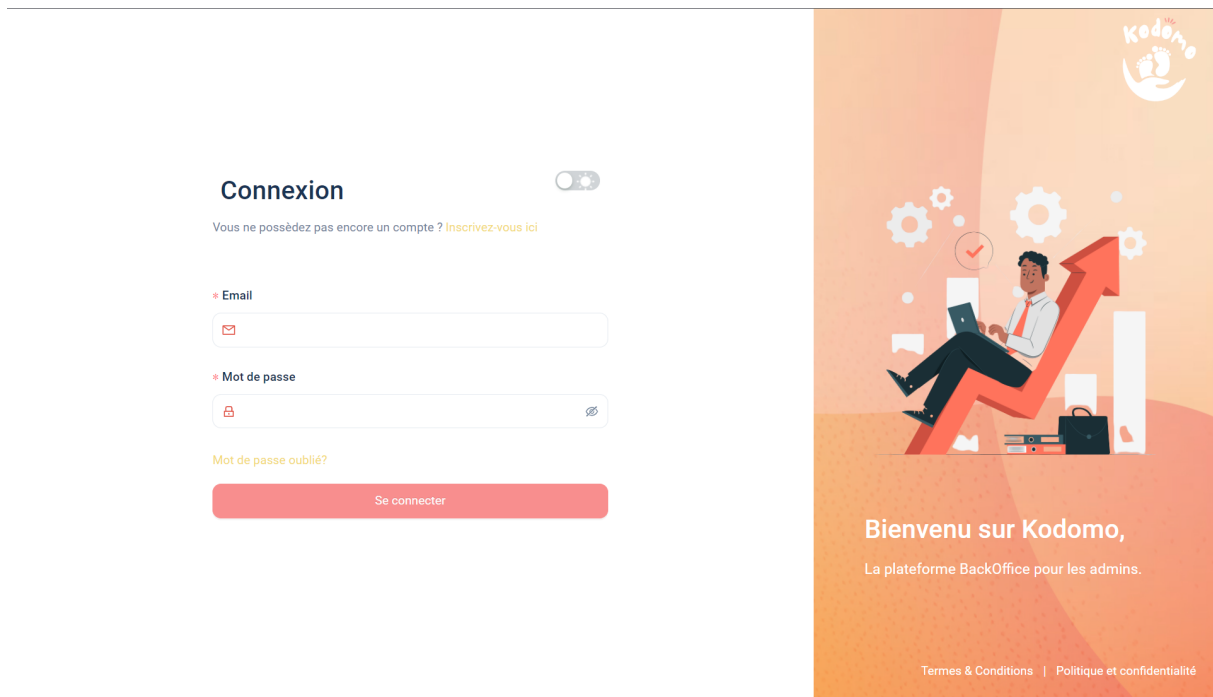
`Gardien(id, email, nom, prenom, password, sexe, type-garde, etat-civile, telephone, codePostale, adresse, nbr-enfant-a-garder, annee-exp, experience-enfant, tarif-par-enfant, evaluation, biography, cv, image, motif-bannissement, statut)`.

## 4.7 Présentation d'interfaces

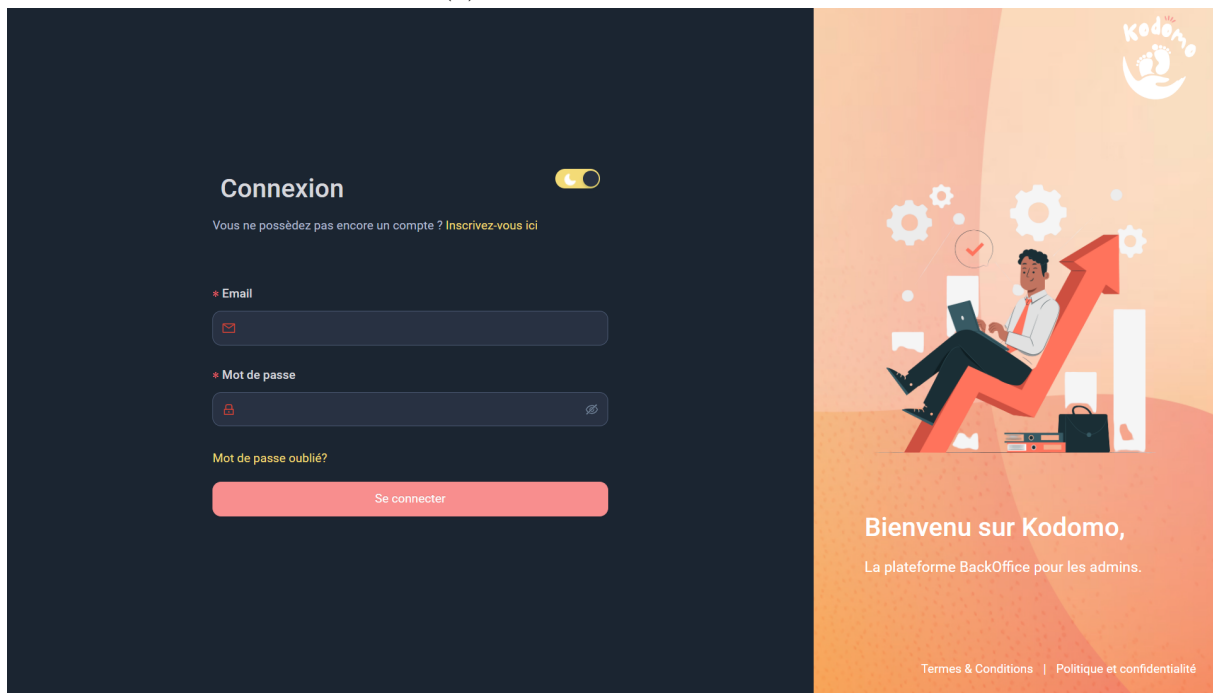
Dans cette section nous allons montrer quelques interfaces de notre plate-forme "KODOMO" relatives au Sprint 1.

### Interface d'authentification

C'est par le biais de cette interface en figure 4.9 que l'utilisateur peut accéder à sa session.



(a) Interface de connexion.



(b) Interface de connexion en mode sombre.

FIG. 4.9 : Interfaces de connexion du back office.

### Interface de gestion des gardiens

La figure 4.10 représente la liste des gardiens qui s'affiche lorsqu'on clique sur la section "Gardiens", le bouton "Ajouter un gardien", nous mène vers le formulaire d'ajout avec toutes les informations nécessaires à ce dernier, notamment son curriculum vitae, le nombre et catégories d'enfants qu'il peut garder, les activités qu'il peut pratiquer avec l'enfant, etc. Comme l'illustre les figures 4.11 et 4.12.

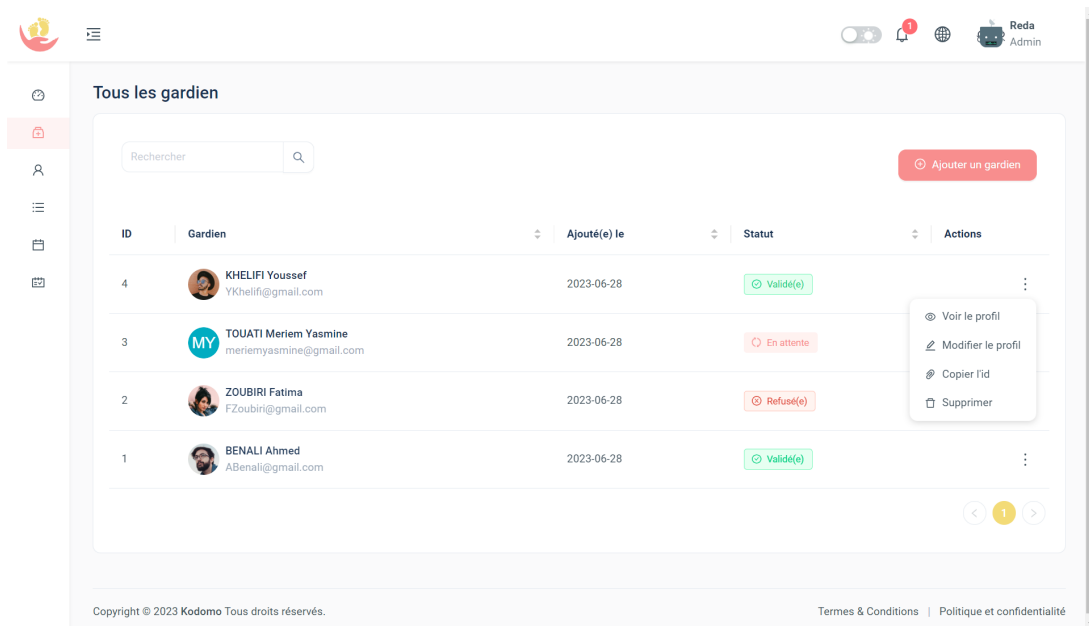


FIG. 4.10 : Interface de gestion des gardiens.

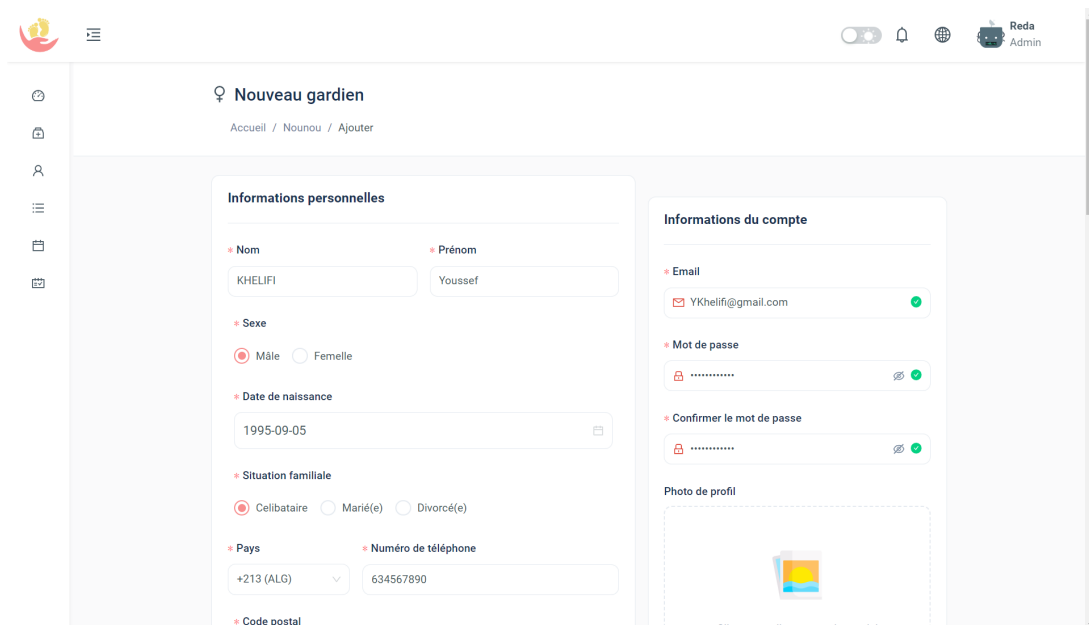
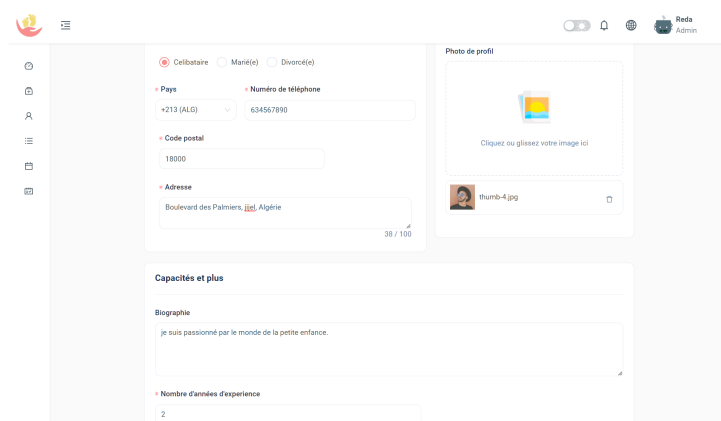
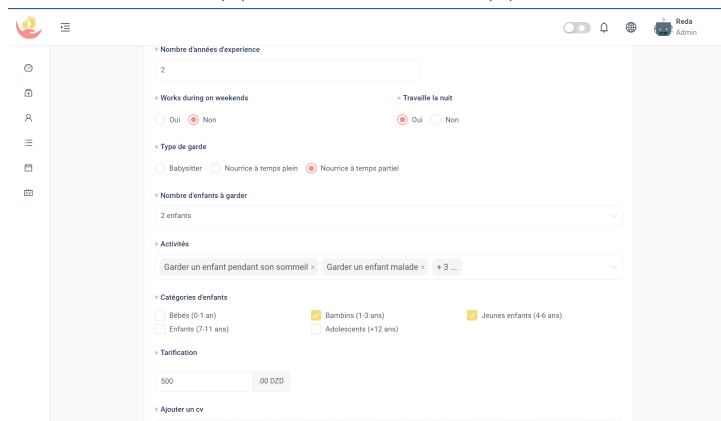


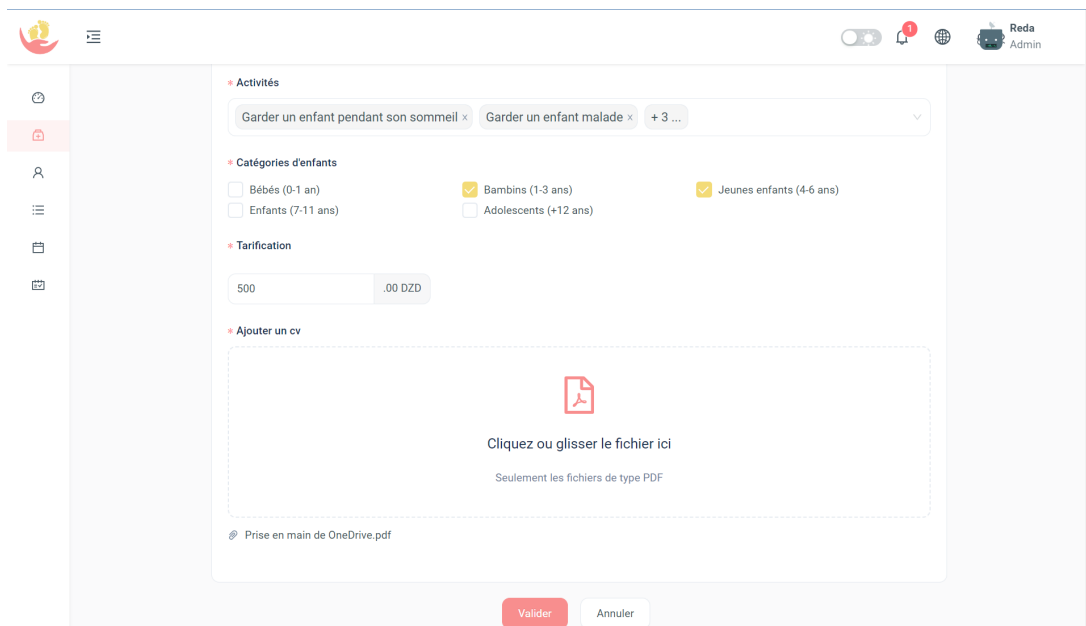
FIG. 4.11 : Interface d'ajout (1).



(a) Interface d'ajout (2).



(b) Interface d'ajout (3).



(c) Interface d'ajout (4).

FIG. 4.12 : Interface du profil d'un gardien.

### 4.8 Conclusion

Le Sprint 1 a été une étape cruciale dans notre processus de développement, nous permettant de mettre en place des fonctionnalités essentielles telles que l'authentification et la gestion des comptes des prestataires c'est à dire les parents et les gardiens. Nous avons également réalisé une série de tests initiaux pour chaque fonctionnalité implémentée pour garantir la qualité de notre code et de notre solution. Dans le prochain chapitre nous déroulerons le sprint 2 et les différentes étapes suivies lors de ce dernier.

# Chapitre 5

## Sprint deux



## 5.1 Introduction

Ce chapitre représente le Sprint 2 du projet, il met en lumière la continuité du travail entamé dans le projet. Ce nouveau sprint représente une étape fondamentale et incontournable pour atteindre nos objectifs. Nous allons nous concentrer sur de nouvelles fonctionnalités et améliorations, en récapitulant les réalisations du sprint précédent.

## 5.2 Backlog du Sprint 2

Ci dessous le 5.1, liste les fonctionnalités que l'on aborde au cours du travail dans cette itération. Nous avons apporté quelques modifications au système suite aux exigences de l'entreprise lors de la réunion du Sprint précédent. Celle-ci a souhaité que nous ajoutions un gestionnaire comme acteur supplémentaire ; nous avons aussi ajouté la partie tableau de bord qui affiche certaines statistiques du système.

TAB. 5.1 : Backlog du Sprint 2.

ID User Story	ID Tâche	Tâches	Durée
US-5	5.1	Élaborer les CU et description de la fonctionnalité "Surveiller l'état du système". Ajouter les diagrammes de séquence et ajouter les classes de conception nécessaires.	3h
	5.2	Développer et tester les fonctionnalités.	30h
US-6	6.1	Élaborer le CU de la fonctionnalité "Administrer un gestionnaire". Ajouter le diagramme de séquence et les classes de conception nécessaires.	7h
	6.2	Développer et tester la fonctionnalité.	48h
US-7	7.1	Élaborer le CU de la fonctionnalité "Gérer les privilèges". Ajouter le diagramme de séquence et les classes de conception nécessaires.	2h
	7.2	Développer et tester la fonctionnalité.	20h

## 5.3 Spécifications

De la même manière que le chapitre précédent (sprint 1), nous passons à l'étape de spécification, pour orienter les étapes ultérieures de conception, développement et test. Nous aborderons :

### 5.3.1 Diagrammes de cas d'utilisation

Le diagramme 5.1 suivant représente la modélisation des cas d'utilisation dérivés du Sprint 2. Les nouveaux cas d'utilisation et les acteurs ajoutés au système sont mis en évidence en couleur afin de mettre en avant les modifications apportées lors de cette itération.

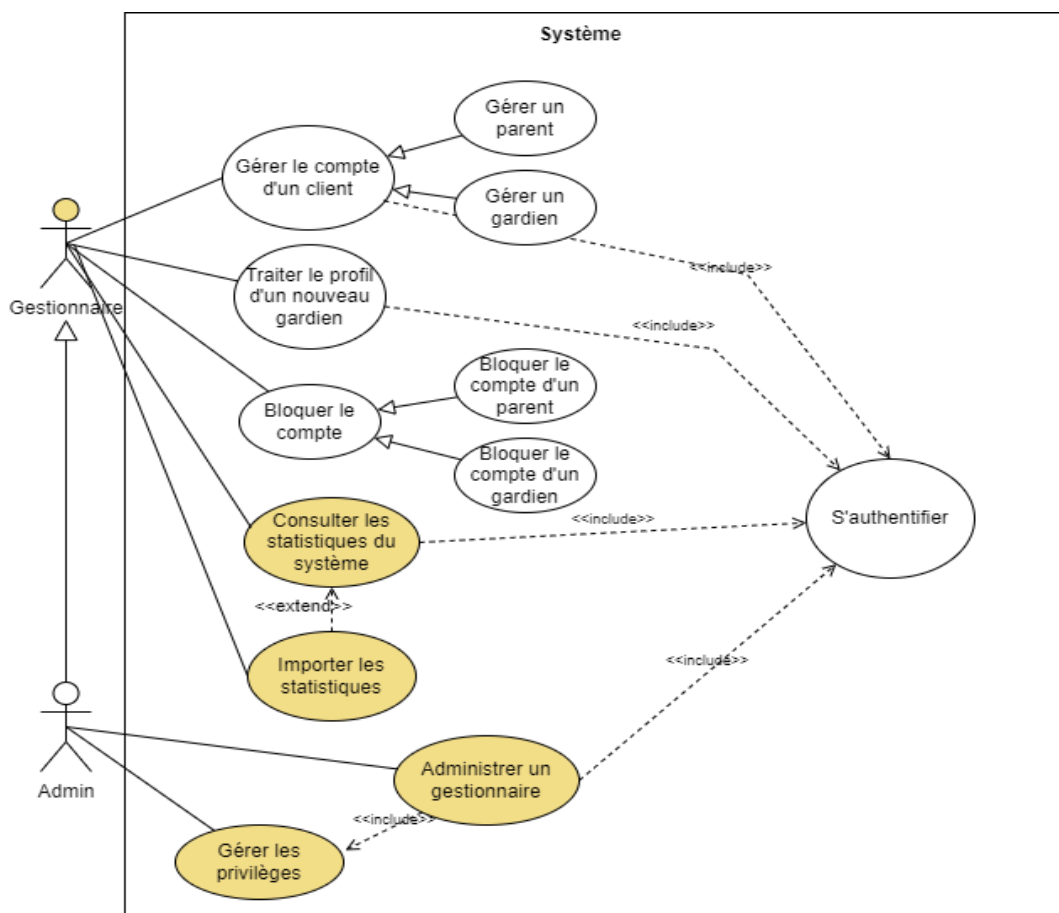


FIG. 5.1 : Diagramme de cas d'utilisation du Sprint 2.

### 5.3.2 Descriptions textuelles

Dans cette section, nous allons présenter la documentation des cas d'utilisation les plus pertinents du Sprint 2, à savoir "Administrer un gestionnaire" et "Consulter les statistiques du système".

### Cas d'utilisation "Administrer un gestionnaire"

TAB. 5.3 : Documentation CU : Administrer un gestionnaire.

Administrer un gestionnaire
<b>ID</b> : 5
<b>Description brève</b> : L'administrateur peut ajouter un gestionnaire au système et gérer ses accès à certaines fonctionnalités (gestions des parents, des gardiens, des offres de garde, etc).
<b>Acteurs primaires</b> : Administrateur
<b>Acteurs secondaires</b> : /
<b>Pré condition</b> : 1. L'administrateur doit être connecté
<b>Enchaînement principal</b> : 1. L'administrateur accède à la section des utilisateurs (gestionnaires et administrateurs). 2. L'application affiche la section avec la liste des gestionnaires existants. A. L'administrateur clique sur le bouton "Ajouter un utilisateur". 3. Le système affiche un formulaire. 4. L'administrateur saisit les informations, puis lui attribue des privilèges à partir de la liste de toutes les fonctionnalités du système. 5. Le système valide les informations. 6. Le système enregistre le nouveau gestionnaire. B. L'administrateur souhaite gérer un gestionnaire existant. 3. L'administrateur sélectionne le gestionnaire souhaité et peut : a. Supprimer le gestionnaire. b. Redéfinir les privilèges du gestionnaire.
<b>Post condition</b> : Le système met à jour les changements effectués.
<b>Enchaînement alternatif</b> : 5.1. Le système affiche que l'utilisateur à ajouter existe déjà dans la base de données.

### Cas d'utilisation "Consulter les statistiques du système"

TAB. 5.5 : Documentation CU : Consulter les statistiques du système.

Consulter les statistiques du système
<b>ID</b> : 6
<b>Description brève</b> : Accéder au tableau de bord pour visualiser des statistiques et des informations sur l'utilisation de l'application.
<b>Acteurs primaires</b> : Gestionnaire
<b>Pré condition</b> : 1. Le gestionnaire doit être connecté
<b>Enchaînement principal</b> : 1. L'application affiche une page d'accueil, qui résume les activités récentes et les statistiques clés de l'application, telles que le nombre total d'utilisateurs, le nombre d'inscriptions, le nombre d'avis et de commentaires, etc. 2. Le gestionnaire peut également filtrer les données selon différents critères. 3. L'application affiche les données filtrées sous forme de graphiques, de tableaux ou de diagrammes, permettant ainsi au gestionnaire de visualiser facilement les tendances et les performances de l'application. 4. Le gestionnaire peut importer les statistiques au format .csv et les lire sur Excel. Il peut utiliser ces données pour prendre des décisions éclairées sur la gestion et l'amélioration de l'application, telles que la mesure de l'efficacité des stratégies de marketing, la surveillance des performances des gardiens, etc.
<b>Post condition</b> : /
<b>Enchaînement alternatif</b> : /

## 5.4 Analyse

Nous allons dans cette phase, utiliser l'ensemble des cas d'utilisation, nous amorcerons la création du modèle du domaine et la conception de quelques diagrammes de séquence système.

### 5.4.1 Modèle de domaine

La figure ci-dessous 5.2 représente la version du modèle de domaine du sprint 2 de notre application.

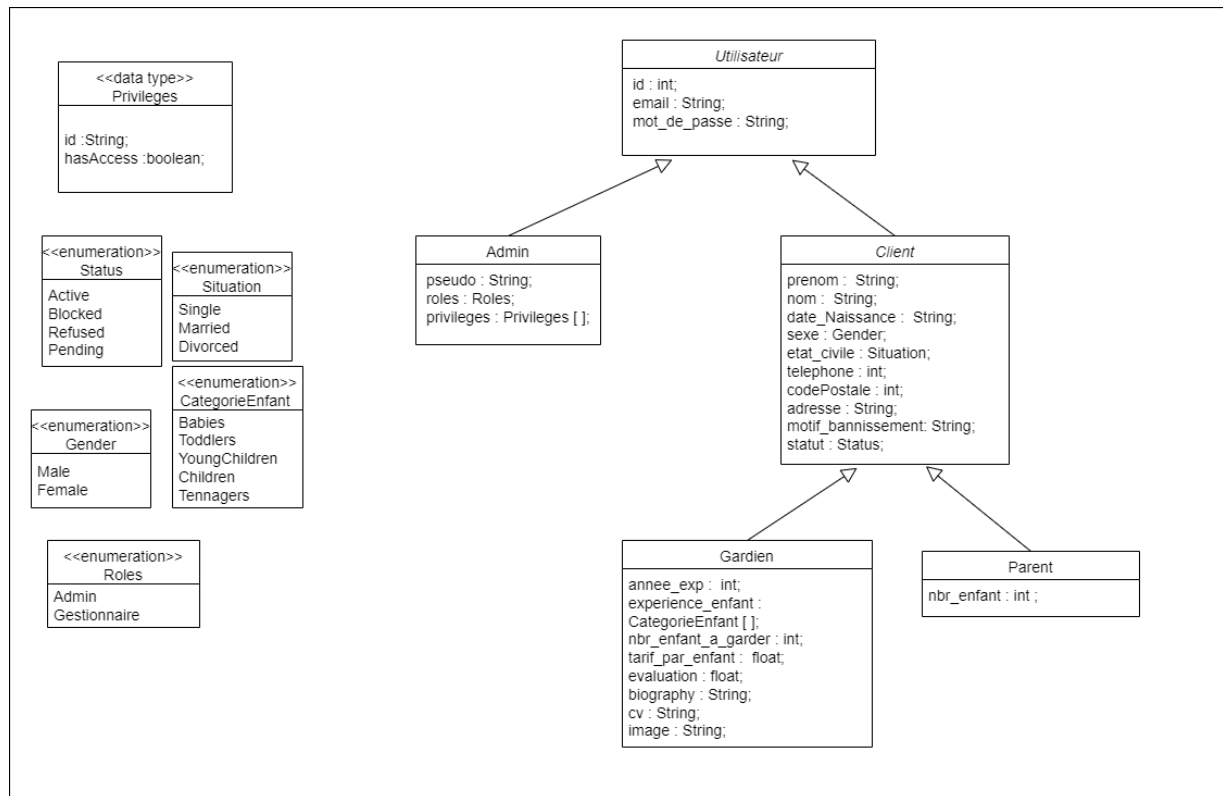


FIG. 5.2 : Modèle de domaine du sprint 2.

### 5.4.2 Diagrammes de séquence système

Dans cette section, nous présenterons les diagrammes de séquence système du deuxième sprint, qui s'intéresse essentiellement aux gestionnaires. Nous mettrons également en évidence les interactions et les étapes clés impliquées dans le processus de gestion de leurs privilèges.

#### Diagramme de séquence système d'administration d'un gestionnaire

Le cas d'utilisation «administrer un gestionnaire» a pour tâche de créer un gestionnaire, définir ses privilèges ou bien de le supprimer. Lors de l'ajout la validation se fait en premier lieu puis une vérification est faite sur l'existence d'un éventuel gestionnaire ayant un même email que celui introduit.

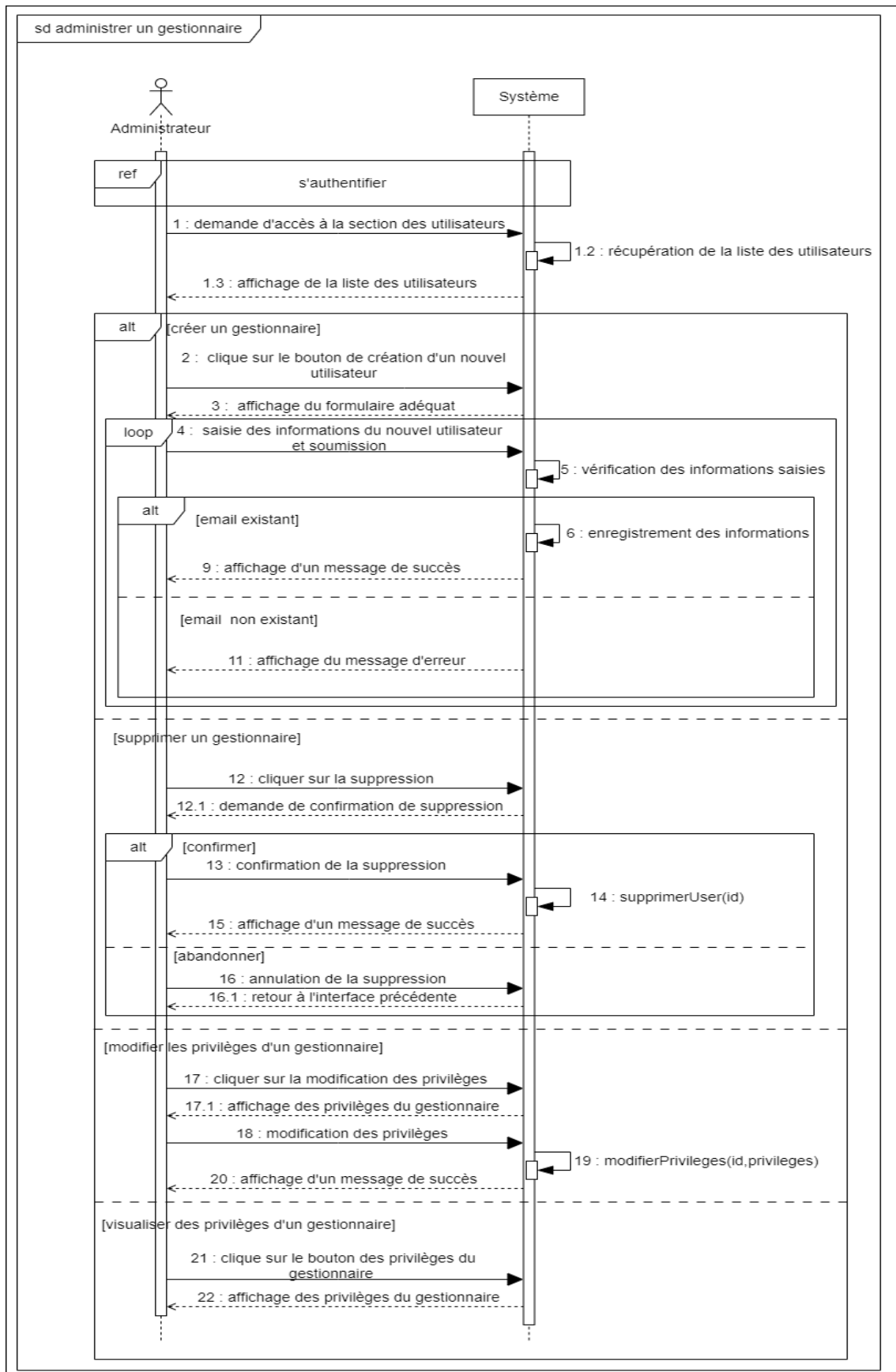


FIG. 5.3 : Diagramme de séquence système du cas d'utilisation «administrer un gestionnaire».

Diagramme de séquence système de traitement d'un profil de gardien

Le cas d'utilisation «traiter le profil d'un gardien» a pour but d'analyser et d'étudier le profil d'un gardien avant de valider sa postulation à l'application Kodomo.

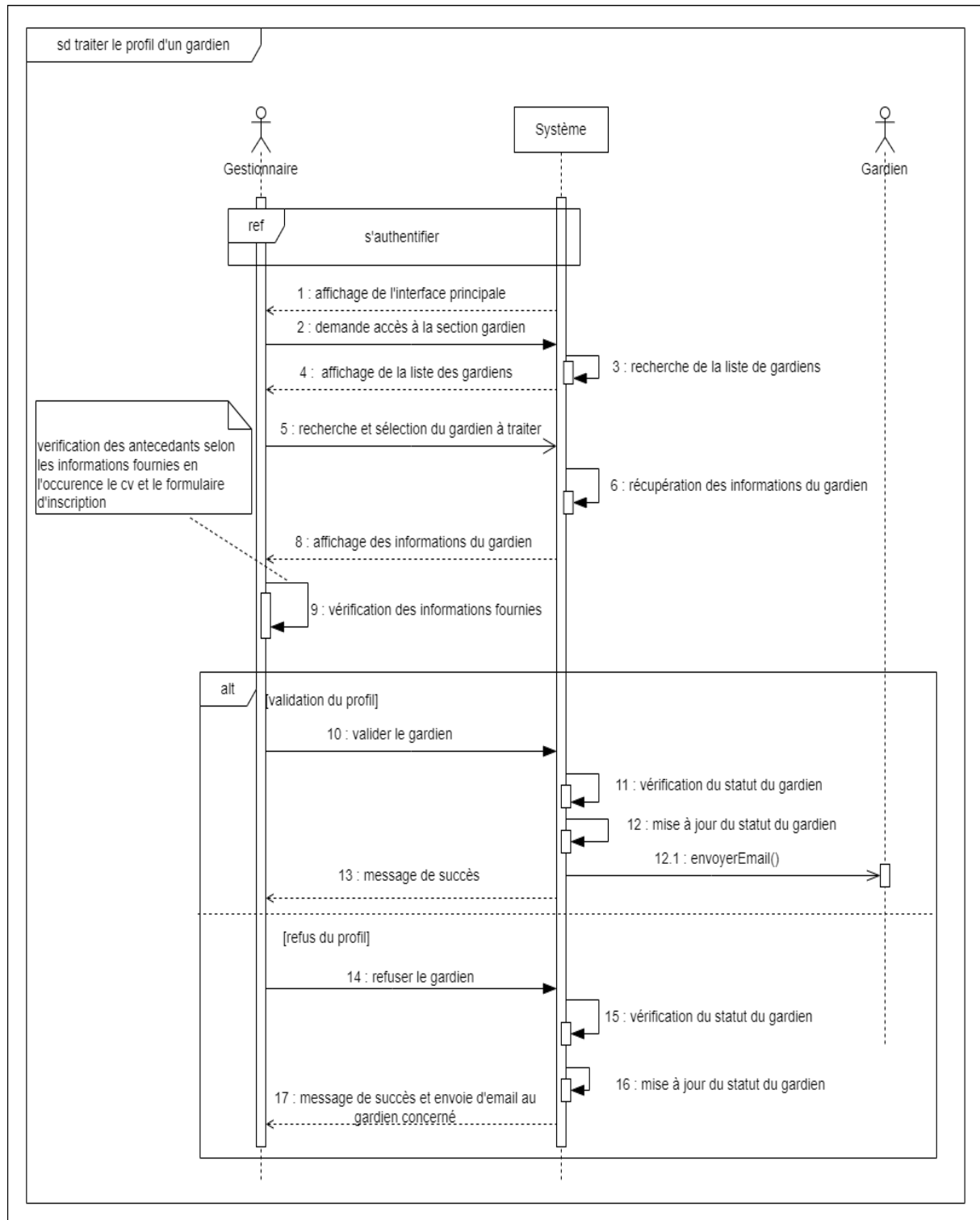


FIG. 5.4 : Diagramme de séquence système du cas d'utilisation «traiter le profil d'un gardien».

## 5.5 Conception

Au cours de cette phase, nous allons réaliser des diagrammes d'activité, de séquence détaillé ainsi qu'un diagramme de classe de conception qu'on peut résumer ainsi :

### 5.5.1 Diagrammes de séquences détaillés

Cette section se concentre sur la présentation du diagramme de séquence détaillé qui concerne l'administration d'un gestionnaire. Ce diagramme offre une vue précise de la façon dont l'administration des gestionnaires est réalisée dans le système, offrant ainsi une compréhension approfondie de cette fonctionnalité essentielle.

#### **Diagramme de séquence détaillé d'administration d'un gestionnaire**

En supplément de ce a qui a été fait dans le diagramme de séquence système (fig.5.3), nous ajouterons l'interaction interne de ses composants.



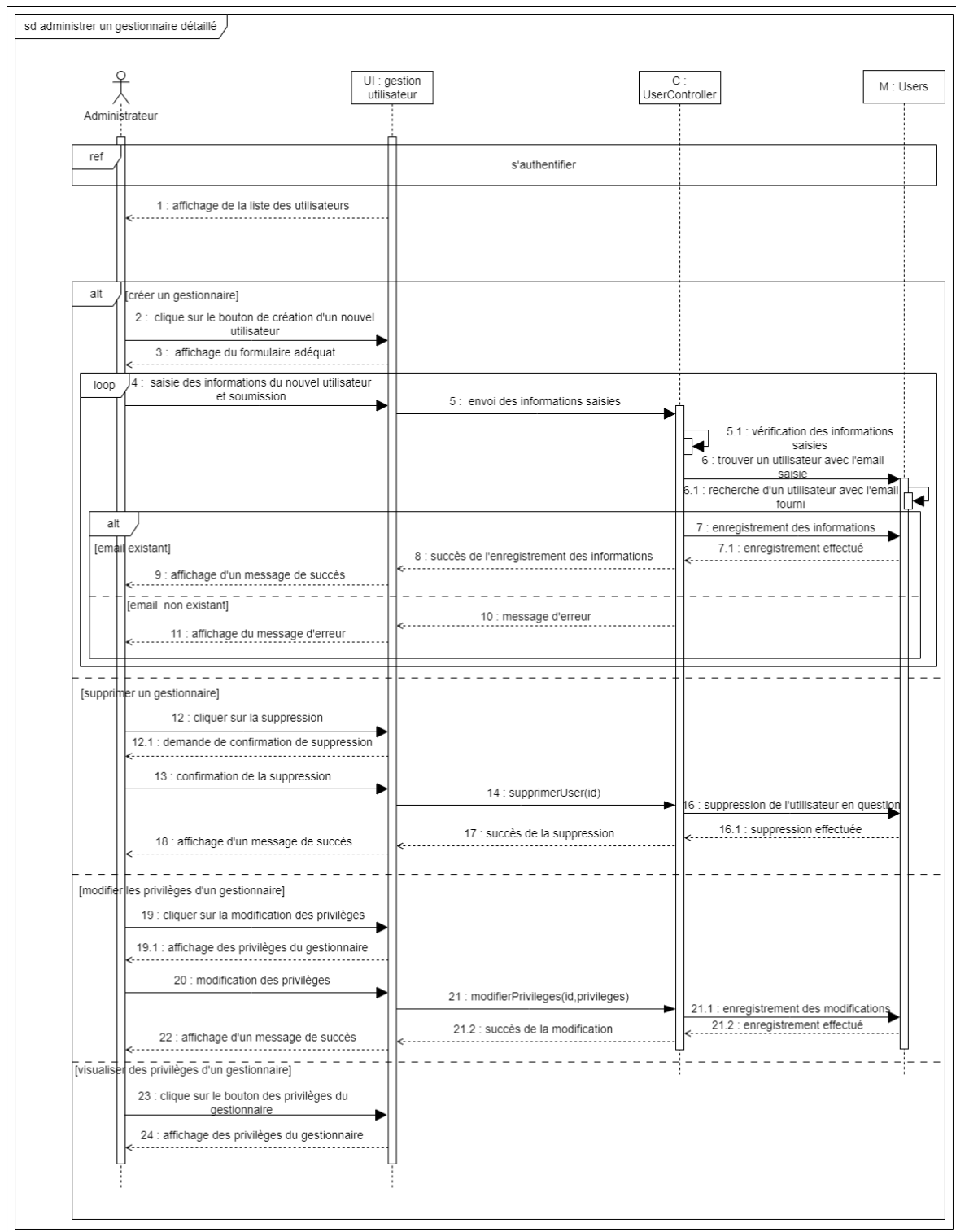


FIG. 5.5 : Diagramme de séquence système du cas d'utilisation «administrer un gestionnaire».

### 5.5.2 Diagrammes de classe de conception

La figure 5.6 représente le diagramme de classe conception du sprint 2 où on peut distinguer l'ajout des fonctionnalités supplémentaires telles que la gestion des privilèges grâce au attributs rôles et privilèges dans la classe "Admin".

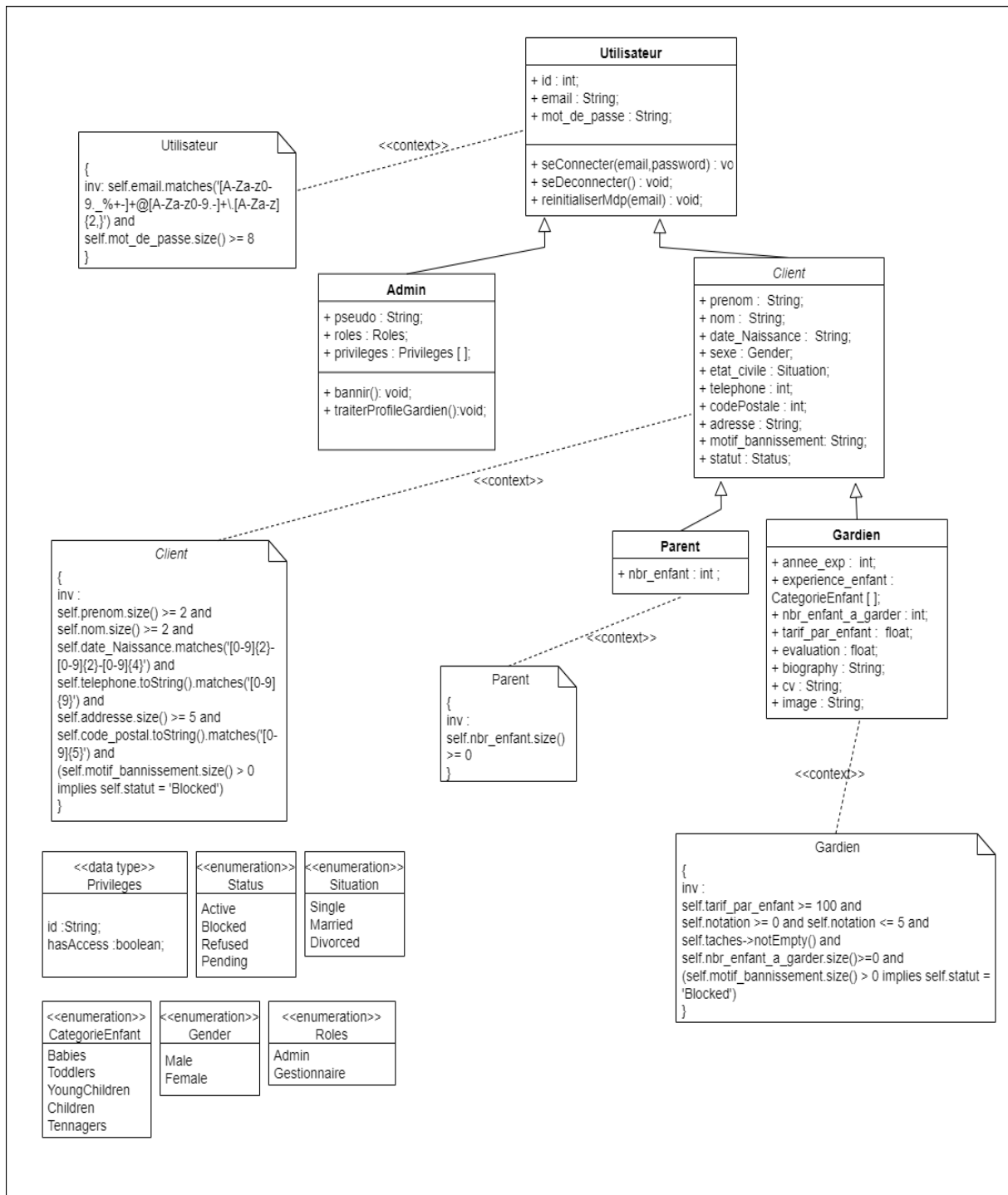


FIG. 5.6 : Diagramme de classe de conception du sprint 2.

## 5.6 Implémentation

Dans cette phase nous allons passer au codage des fonctionnalités définies, en commençant par le modèle relationnel de données.

### 5.6.1 Modèle relationnel de données

Ce dernier est presque similaire à celui que nous avons défini lors du Sprint précédent, la différence entre les deux schémas est donc minime, nous avons ajouté les attributs "roles" et "privileges".

Admin(id, pseudo, email, password, roles, privileges).

Parent(id, email, nom, prenom, password, sexe, etat-civile, telephone, codePostale, adresse, nbr-enfant, motif-bannissement, statut).

Gardien(id, email, nom, prenom, password, sexe, type-garde, etat-civile, telephone, codePostale, adresse, nbr-enfant-a-garder, annee-exp, experience-enfant, tarif-par-enfant, evaluation, biography, cv, image, motif-bannissement, statut).

## 5.7 Présentation d'interfaces

Dans cette section nous allons montrer quelques interfaces de notre plate-forme "KODOMO" relatives au Sprint 2.

### Interface des gestionnaires et des administrateurs

Quand on clique sur la section "Gérer les utilisateurs", le système affiche la liste de ces derniers. Pour chacun, on peut voir les autorisations qu'ils possèdent dans le système, en déroulant la petite partie intitulé "Voir les privilèges", on peut aussi modifier les privilèges d'un gestionnaire, figure 5.8b.

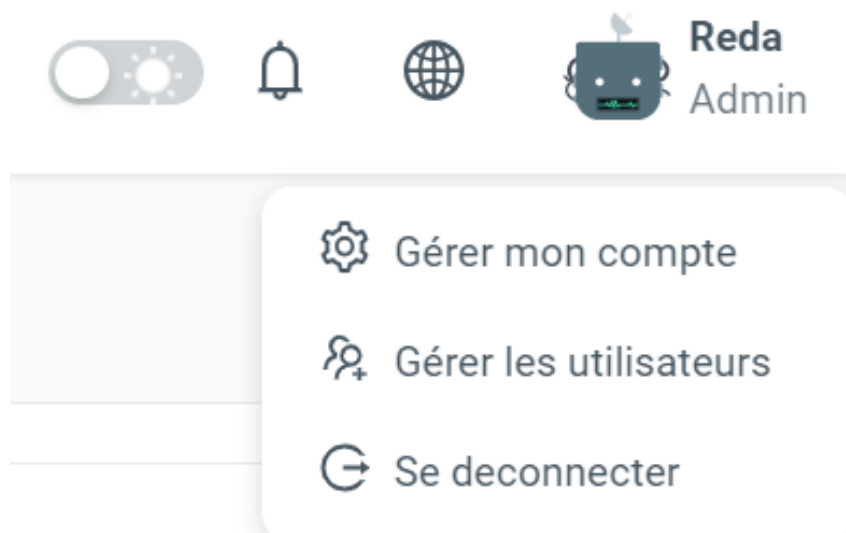
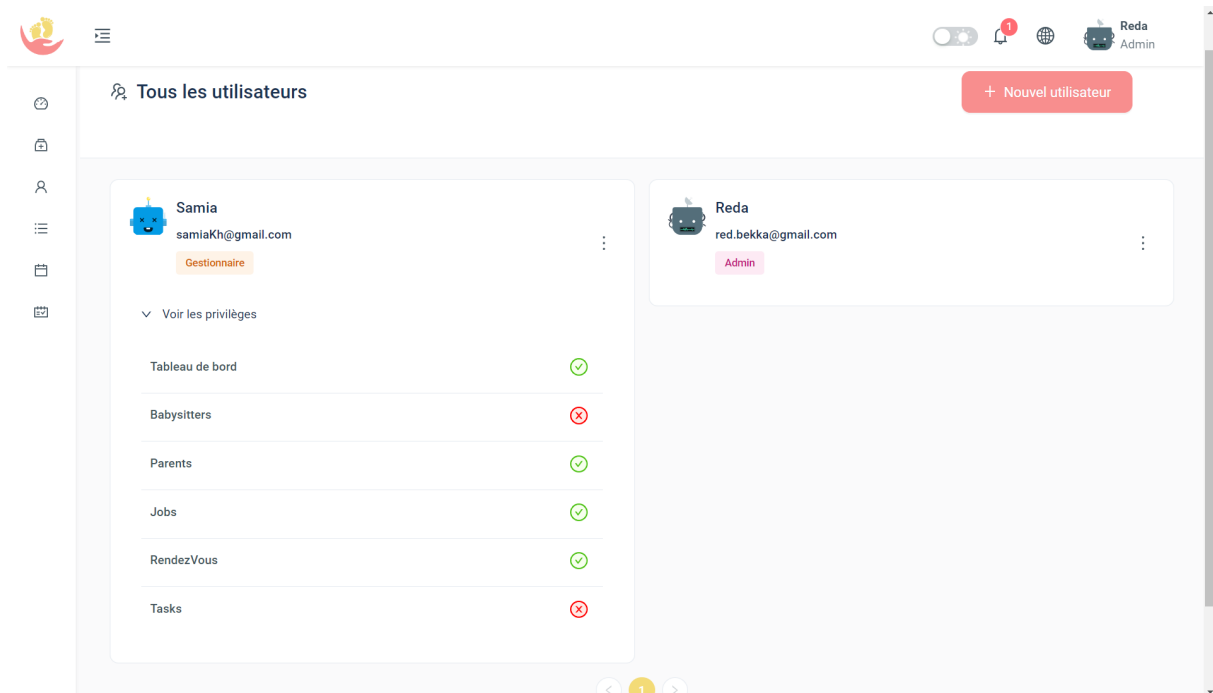
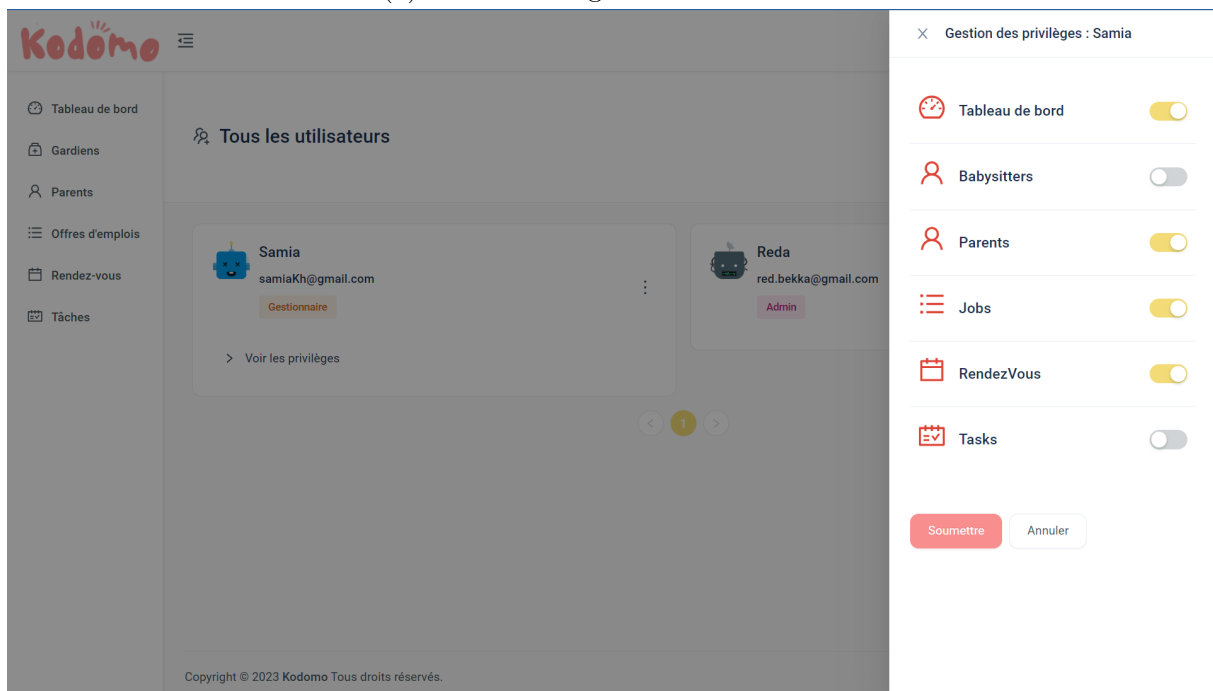


FIG. 5.7 : Section de gestion tous les admins.



(a) Interface de gestion des Admins.



(b) Privilèges d'un des gestionnaires.

FIG. 5.8 : Interface des gestionnaires et administrateurs.

### Interface du tableau de bord

Après l'authentification, nous sommes redirigées vers cette page ; c'est ici qu'on peut visionner les statistiques du système, telles que le nombre de parents inscrits, le nombre de gardiens enregistrés dans le système, le nombre de gardiens actifs, bloqués ou refusés. On a aussi un graphique qui répartit les gardiens évalués par les parents en terme de note. Pour télécharger les statistiques en fichier .csv , il est mis à disposition par un simple clique sur le bouton "Importer les résultats".

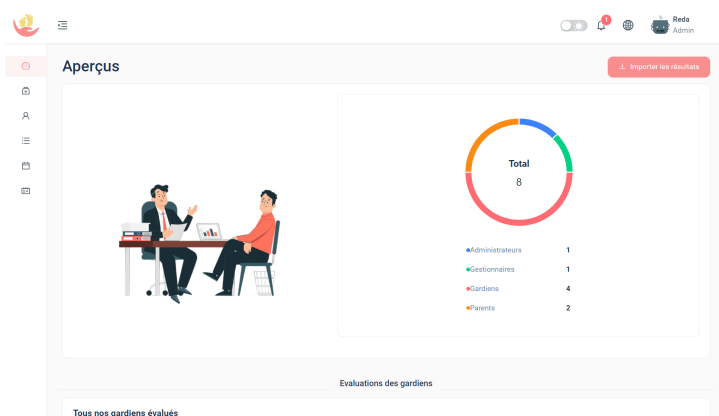


FIG. 5.9 : Interface du tableau de bord (1).



FIG. 5.10 : Interface du tableau de bord sombre(2).

## 5.8 Conclusion

Dans le sprint 2, une des fonctionnalités majeures que nous avons ajoutées est l'administration des gestionnaires et de leurs privilèges. Cela nous permet de définir différents niveaux d'accès et de contrôle au sein du système, en attribuant des droits spécifiques à chaque gestionnaire en fonction de son rôle et de ses responsabilités. L'un des principaux aspects du prochain sprint sera la mise en œuvre des fonctionnalités essentielles qui compléteront le système. Nous accorderons une attention particulière à la gestion des offres de garde, à la postulation à ces offres et à l'organisation des rendez-vous pour les entretiens parent-gardien.

# Chapitre 6

## Sprint trois

## 6.1 Introduction

Dans ce chapitre, nous aborderons en détail le sprint 3 de notre projet. Après avoir parachevé les sprints 1 et 2, le sprint 3 représente une étape importante de notre progression. Ce chapitre mettra en évidence les tâches réalisées, les défis rencontrés et les résultats obtenus lors de ce sprint.

## 6.2 Backlog du Sprint 3

Résumé dans le tableau 6.1 ci-dessous.

TAB. 6.1 : Backlog du Sprint 3.

ID User Story	ID Tâches	Tâches	Durée
US-8	8.1	Élaborer les CU et description de la fonctionnalité "Créer un tâche de garde". Ajouter les diagrammes de séquence et ajouter les classes de conception nécessaires.	2h
	8.2	Développer et tester les fonctionnalités.	10h
US-9	9.1	Élaborer les CU et description de la fonctionnalité "Gérer les entretiens de garde". Ajouter les diagrammes de séquence et ajouter les classes de conception nécessaires.	6h
	9.2	Développer et tester les fonctionnalités.	45h
US-10	10.1	Élaborer le CU de la fonctionnalité "Créer une offre de garde". Ajouter le diagramme de séquence et les classes de conception nécessaires.	7h
	10.2	Développer et tester la fonctionnalité.	52h
US-11	11.1	Élaborer le CU de la fonctionnalité "Initier une demande d'entretien". Ajouter le diagramme de séquence et les classes de conception nécessaires.	4h
	11.2	Développer et tester la fonctionnalité.	15h

Table 6.1 Backlog du Sprint 3. (suite)

ID User Story	ID Tâches	Tâches	Durée
US-12	12.1	Élaborer le CU de la fonctionnalité "Postuler à une offre de garde". Ajouter le diagramme de séquence et les classes de conception nécessaires.	4h
	12.2	Développer et tester la fonctionnalité.	15h

### 6.3 Spécification

Dans cette partie, de même pour les sprints précédents, nous abordons les spécifications relatives à ce sprint.

#### 6.3.1 Diagrammes des cas d'utilisation

Nous regroupons ici la représentation de toutes les fonctionnalités citées dans le backlog du sprint 3 figure 6.1.



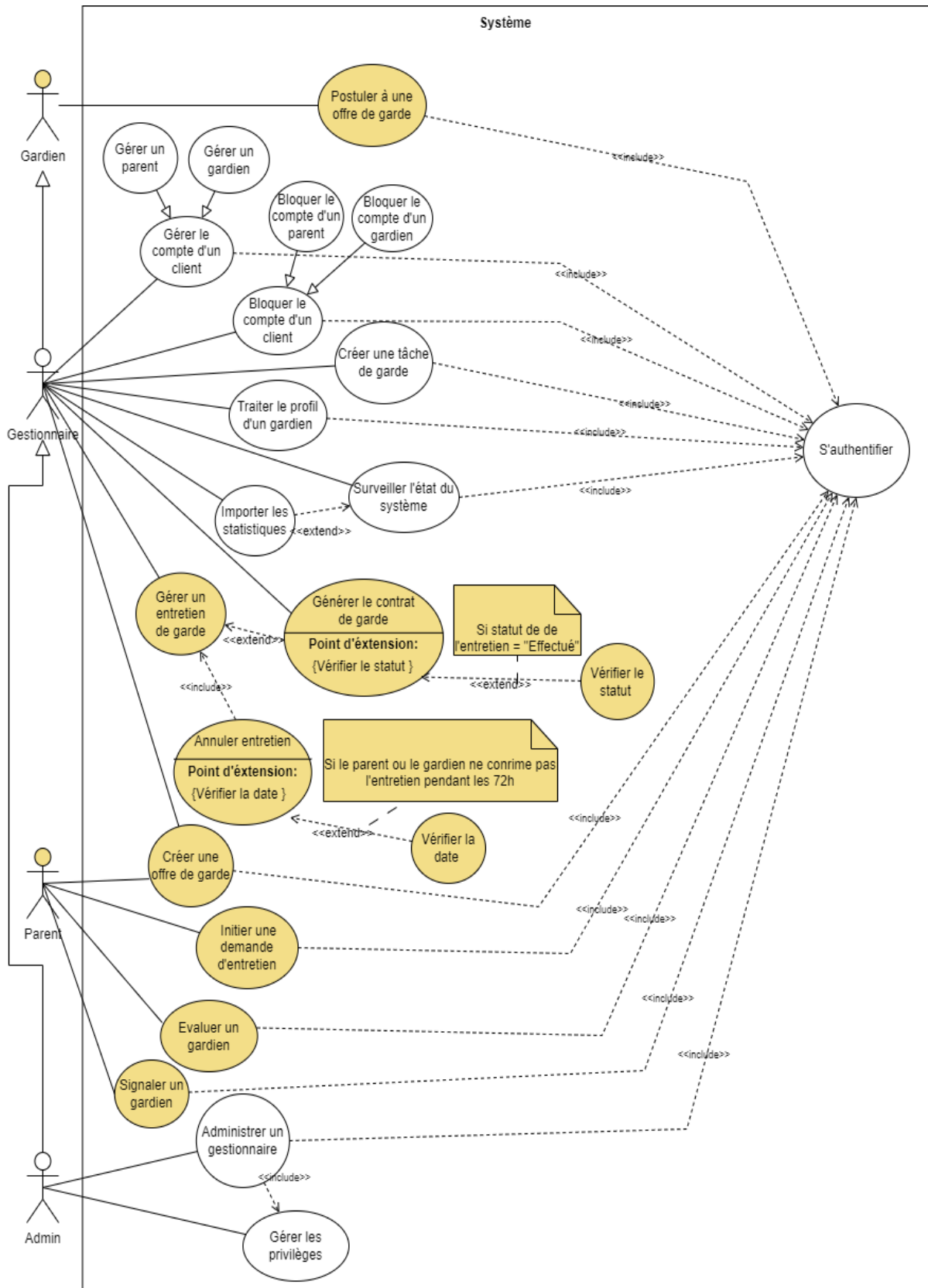


FIG. 6.1 : Diagramme de cas d'utilisation du Sprint 3.

### 6.3.2 Descriptions textuelles

Nous présentons ici les cas d'utilisation (créer une offre de garde, gérer un entretien, postuler à une offre de garde et créer une tâche) illustrés dans les tableaux qui suivent :

#### Cas d'utilisation "Créer une offre de garde"

TAB. 6.2 : Règles de gestion : Créer une offre de garde.

ID	Règles de gestion
RG-008	Le nombre d'enfants à garder doit être supérieur ou égal à 0 et inférieur ou égal à 4 <sup>1</sup> .
RG-009	Le type de garde doit être soit désigné en "Babysitter", ou en "Nourrice à temps partiel", ou en "Nourrice à temps plein".
RG-010	Dans le cas d'un choix d'une nourrice à temps partiel, le temps de travail saisi doit être inférieur à 48h par semaine [12].
RG-011	Dans le cas d'un choix d'une nourrice à temps plein, le temps de travail saisi ne doit pas dépasser les 48h par semaine [5].

TAB. 6.3 : Documentation CU : Créer une offre de garde.

<b>CU</b> : Créer une offre de garde
<b>ID</b> : 7
<b>Description brève</b> : L'opération permet à un parent d'annoncer ses offres aux gardiens disponibles dans l'application.
<b>Acteurs primaires</b> : Parent, gestionnaire
<b>Pré condition</b> : L'acteur est déjà connecté
<p><b>Enchaînement principal</b> :</p> <p>Le cas d'utilisation démarre lorsque l'acteur souhaite trouver un gardien pour garder son enfant.</p> <ol style="list-style-type: none"> <li>1. L'acteur accède à la section de création d'une offre dans l'application.</li> <li>2. L'application affiche un formulaire de demande de garde avec les champs nécessaires à saisir.</li> <li>3. L'acteur effectue la saisie avec les informations concernant la garde et envoie sa demande.</li> </ol>

4. L'application l'informe que son offre a bien été ajoutée.
<b>Post condition</b> : L'acteur attend de recevoir une demande de candidature.
<b>Enchaînement alternatif</b> : /

**Cas d'utilisation "Gérer un entretien"**

TAB. 6.4 : Règles de gestion : Gérer un entretien entre un parent et un gardien.

ID	Règles de gestion
RG-012	La date de l'entretien doit être supérieure à la date courante.

TAB. 6.5 : Documentation CU : Gérer un entretien entre un parent et un gardien.

Gérer un entretien entre un parent et un gardien
<b>ID</b> : 8
<b>Description brève</b> : Cette opération permet à l'administrateur de gérer les statuts d'un rendez-vous pour un entretien physique entre la nourrice candidate et le parent ayant accepté sa candidature.
<b>Acteurs primaires</b> : Gestionnaire
<b>Acteurs secondaires</b> : Parent, nourrice (gardien)
<b>Pré-condition</b> : 1. Le gestionnaire doit être connecté
<p><b>Enchaînement principal</b> :</p> <p>Le cas d'utilisation démarre lorsque l'administrateur est notifié qu'une demande de rendez-vous a été faite par un parent :</p> <ol style="list-style-type: none"> <li>1. Le gestionnaire accède à la section des rendez-vous.</li> <li>2. L'application affiche la liste de tous les rendez-vous.</li> <li>3. Le gestionnaire sélectionne le plus récent.</li> <li>4. L'application affiche un formulaire lui indiquant de saisir la date et l'heure de l'entretien.</li> <li>5. Le gestionnaire effectue la saisie et valide son choix.</li> <li>6. L'application envoie par la suite un e-mail aux deux parties concernées, soit le parent et le gardien, le statut du rendez-vous devient "en attente de confirmation".</li> <li>7. Les deux parties doivent par la suite confirmer la réception de l'e-mail.</li> </ol>

<b>Post condition</b> : Le statut du rendez-vous devient donc "confirmé".
<b>Enchaînement alternatif</b> :
7.1. Si l'une des parties ne valide pas le rendez-vous durant les 72 heures qui suivent l'envoi, le statut devient "Annulé".

Cas d'utilisation "Postuler à une offre de garde"

TAB. 6.6 : Règles de gestion : Postuler à une offre de garde.

ID	Règles de gestion
RG-013	Le gardien ne peut postuler qu'aux offres dont le type de garde et certains autres critères correspondent à son profil <sup>2</sup> .

TAB. 6.7 : Documentation CU : Postuler à une offre de garde.

<b>CU</b> : Postuler à une offre de garde
<b>ID</b> : 9
<b>Description brève</b> : Cette opération permet au gardien d'envoyer sa candidature au parent ayant créé une offre de garde.
<b>Acteurs primaires</b> : Gardien
<b>Acteurs secondaires</b> : Parent
<b>Pré-condition</b> : Le gardien est déjà connecté
<b>Enchaînement principal</b> :
<ol style="list-style-type: none"> <li>1. Le gardien accède à la section des offres de garde.</li> <li>2. L'application affiche la liste des offres de garde.</li> <li>3. Le gardien peut choisir de faciliter sa recherche en filtrant la liste selon les critères qu'il souhaite avoir.</li> <li>4. Le gardien choisit une offre et clique sur le bouton "Postuler à cette offre".</li> <li>5. L'application envoie et notifie ensuite le parent de la candidature.</li> </ol>
<b>Post condition</b> : La candidature prend le statut "en attente", et le gardien attend une réponse.

**Enchaînement alternatif** : Le gardien peut se rétracter et annuler la candidature à condition que celle-ci ne soit pas encore validée.

### Cas d'utilisation "Créer une tâche de garde"

TAB. 6.8 : Documentation CU : Créer une tâche de garde.

<b>CU</b> : Créer une tâche de garde
<b>ID</b> : 10
<b>Description brève</b> : Cette opération permet à l'administrateur/gestionnaire d'ajouter de nouvelles tâches au système que les gardiens pourront effectuer et que les parents peuvent choisir lors de la création d'une offre de garde.
<b>Acteurs primaires</b> : Gestionnaire
<b>Acteurs secondaires</b> : /
<b>Pré-condition</b> :
1. Le gestionnaire doit être déjà connecté.
<b>Enchaînement principal</b> :
1. Le gestionnaire accède à la section des tâches.
2. L'application affiche la liste des tâches.
3. Le gestionnaire clique sur "Ajouter une nouvelle tâche".
4. L'application le mène à un formulaire.
5. Le gestionnaire saisit le libellé de la tâche et peut choisir de lui assigner une image indicative, puis il valide son choix.
6. L'application affiche un message de succès.
<b>Post condition</b> : La tâche s'ajoute à la base de données.
<b>Enchaînement alternatif</b> : /

## 6.4 Analyse

Suite aux sprints précédents, et à l'apparition de nouvelles exigences inhérente à l'élaboration du sprint 3, nous avons pu dégager des fonctionnalités supplémentaires nécessaires au bon fonctionnement de la plate-forme, et que nous illustrons selon les prochaines étapes.

### 6.4.1 Modèle de domaine

Le modèle de domaine de ce sprint, indiquera les entités intervenantes après l'analyse. A noter que par manque d'espace on a omis quelques éléments définis dans les sprints précédents.

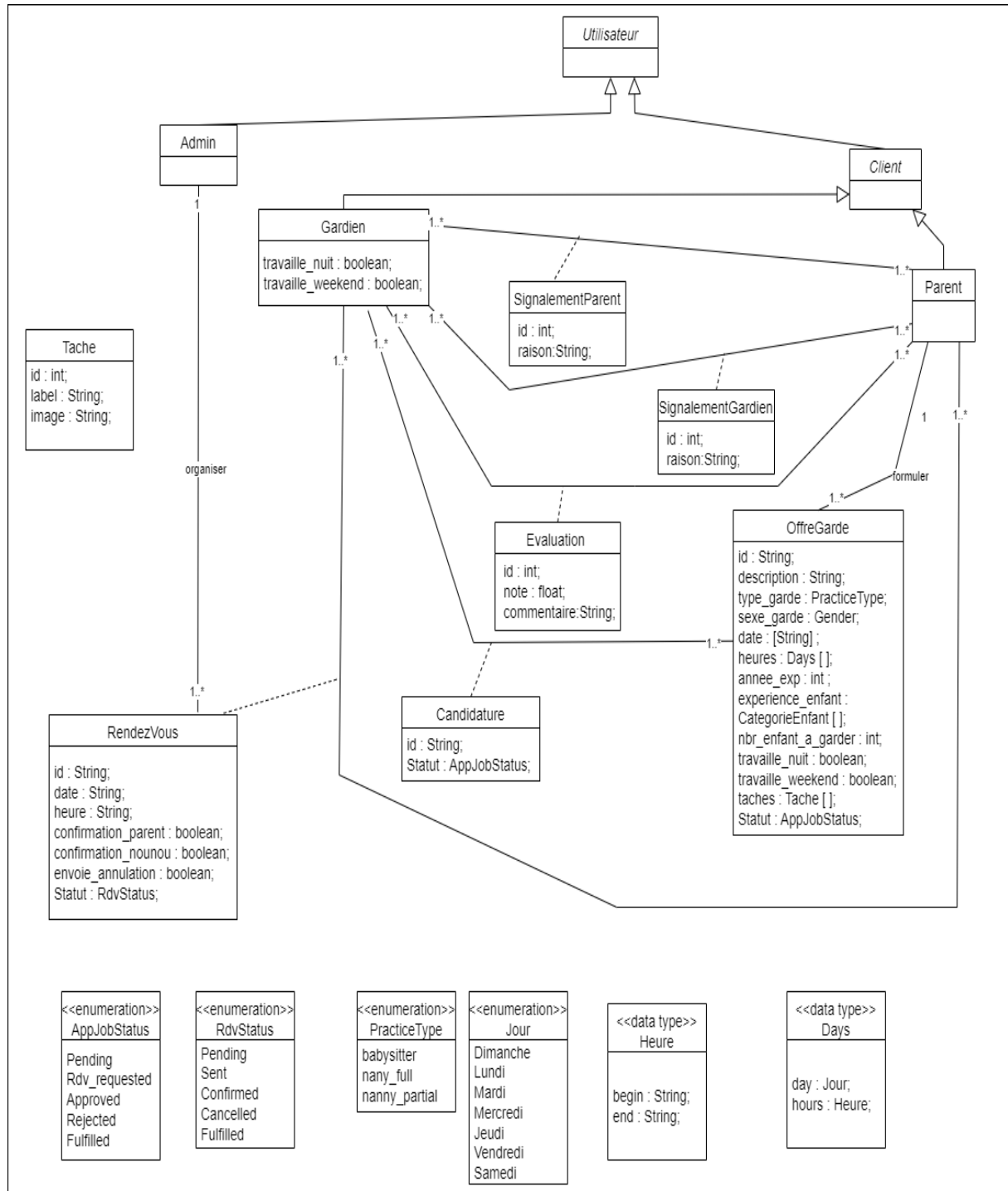


FIG. 6.2 : Modèle de domaine du sprint 3

### 6.4.2 Diagrammes de séquence système

Nous nous intéresserons qu'aux cas pertinents de ce sprint à savoir «créer une offre de garde», «postuler à une offre de garde» et «gérer un entretien ».

#### Diagramme de séquence système de création d'une offre de garde

Le cas d'utilisation «créer une offre de garde» a pour but de formuler une offre de garde créée par un parent. Dans le cas d'une quelconque erreur de saisie le système la détecte puis permet au parent de ressaisir les bonnes informations dans les champs erronés.

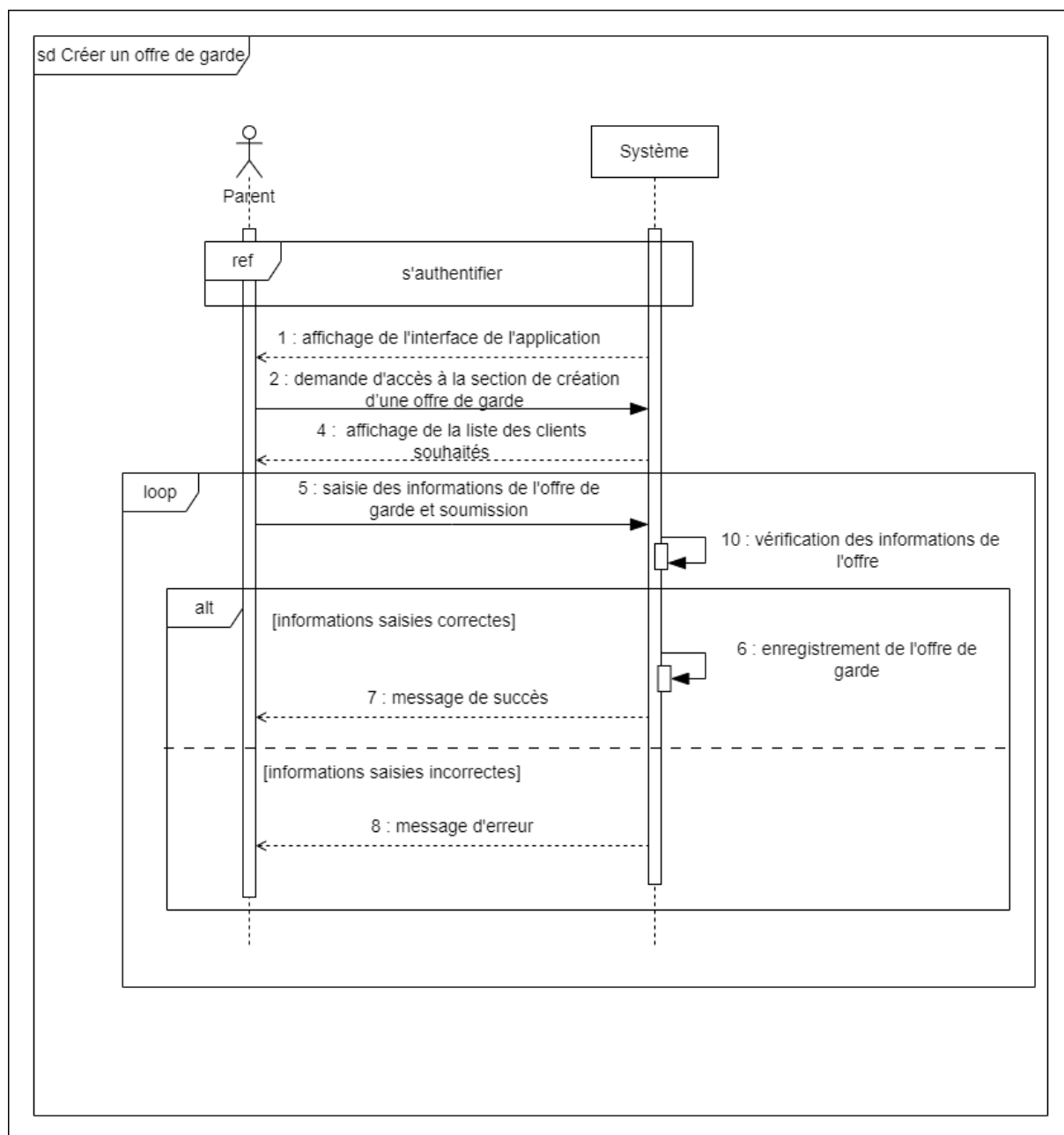


FIG. 6.3 : Diagramme de séquence système du cas d'utilisation «créer une offre de garde».

Diagramme de séquence système de candidature à une offre de garde

Le cas d'utilisation «postuler à une offre de garde» donne la possibilité à une nourrice de postuler à une offre de garde, si celle-ci répond aux critères listés dans l'offre de garde.

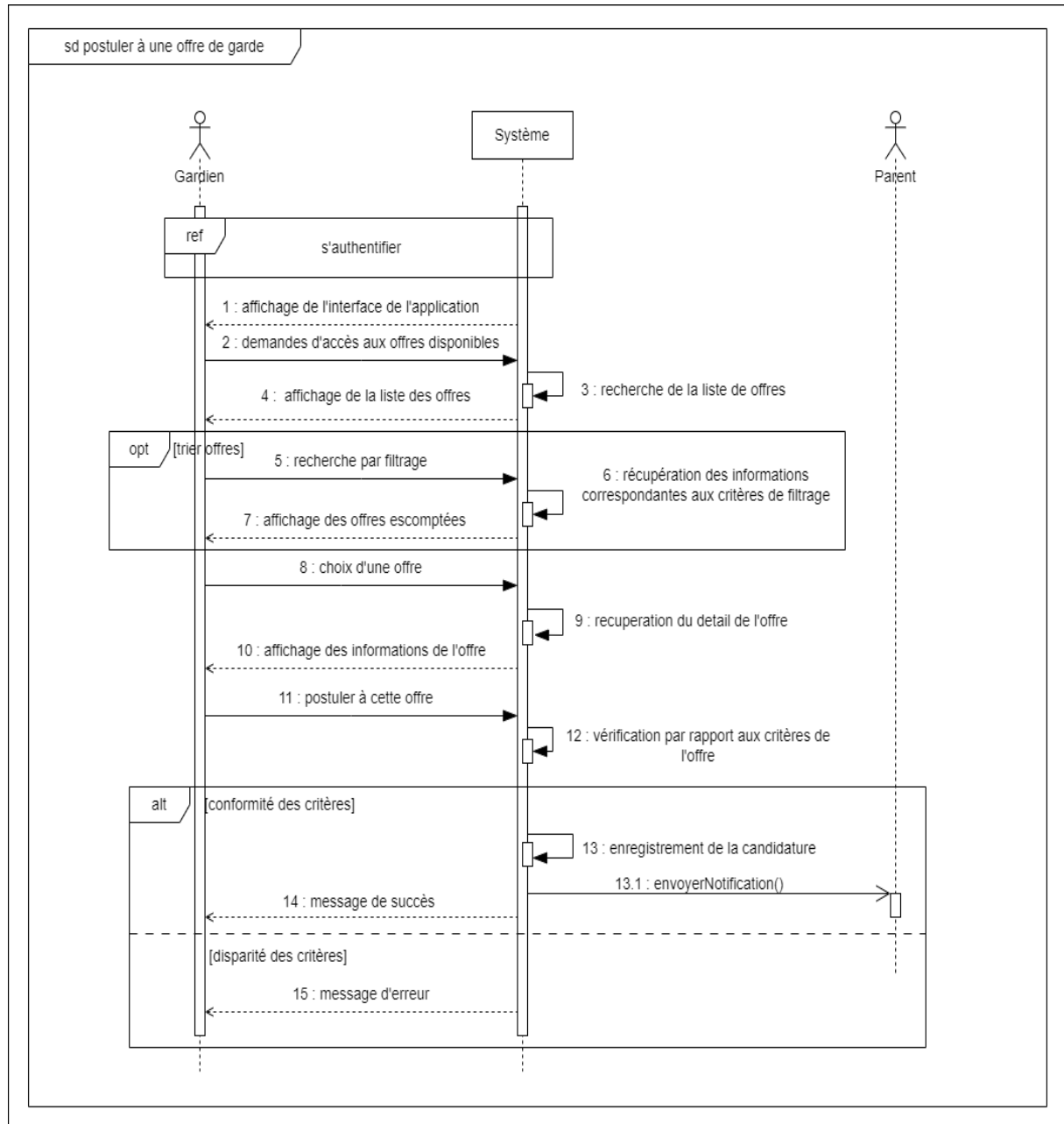


FIG. 6.4 : Diagramme de séquence système du cas d'utilisation «postuler à une offre de garde».

Diagramme de séquence système de gestion d'un entretien

Le cas d'utilisation «Gérer un entretien» fixe un rendez-vous pour une offre donnée dans le but d'une entrevue entre les parties concernées. Plusieurs scénarios peuvent surgir,



le plus intéressant est celui d'annulation dans le cas où les parties prenantes ne respectent pas les délais de confirmation. Voir figure 6.5.

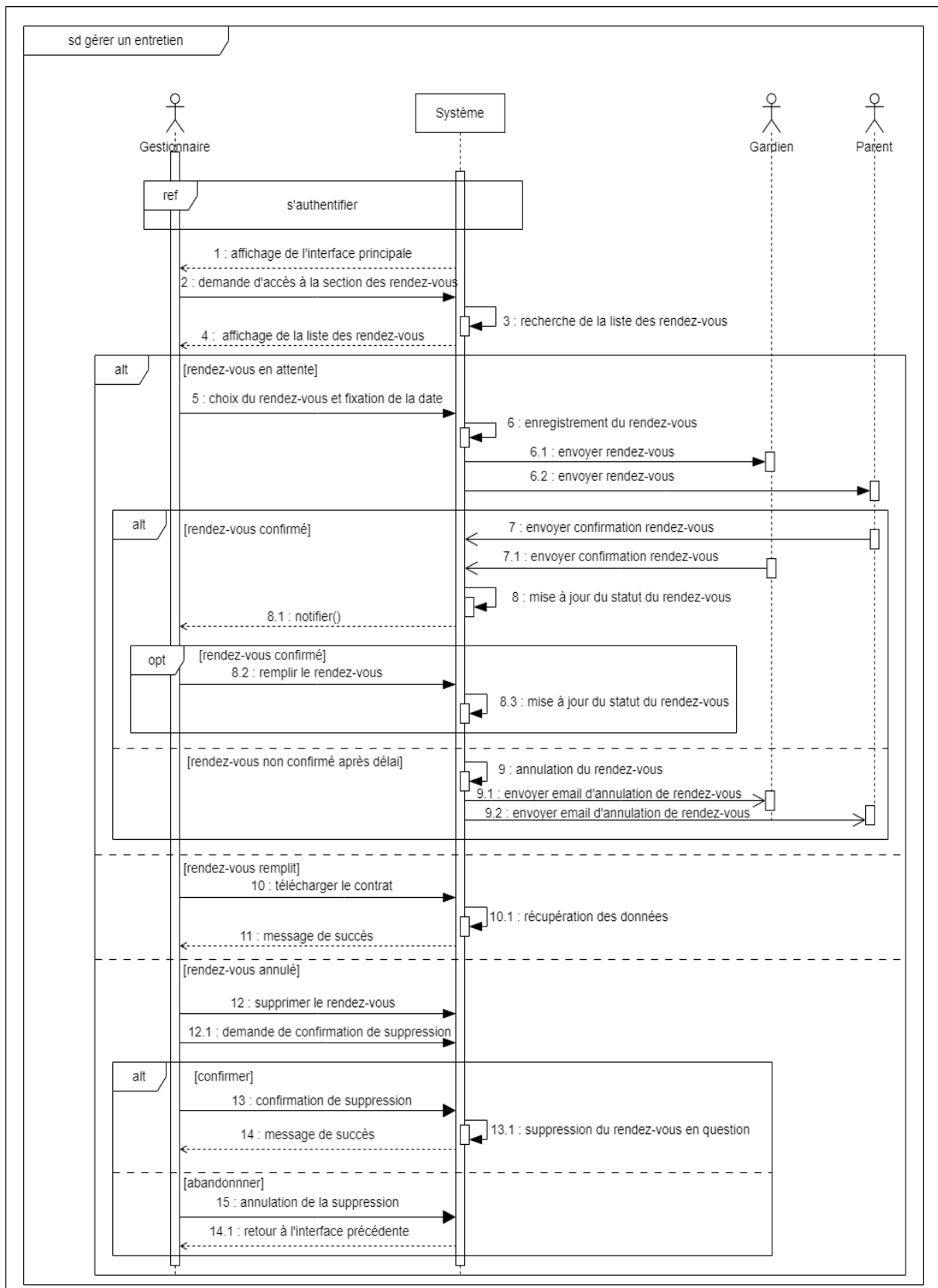


FIG. 6.5 : Diagramme de séquence système du cas d'utilisation «Gérer un entretien».

### 6.5 Conception

Après réalisation des diagrammes de séquences et du modèle de domaine, nous sommes parvenues à détailler l'enchaînement de quelques aspects pertinents de la plate-forme, les diagrammes qui suivent sont un complément aux informations avancées plutôt.

#### 6.5.1 Diagrammes de séquences détaillés

Nous nous intéressons dès à présent à analyser la communication et le comportement dynamique d'un système lors du troisième sprint. On notera : M : Modèle, UI : interface utilisateur, C : Contrôleur, suivant la subdivision :

##### Diagramme de séquence détaillé de création d'une offre de garde

En plus du diagramme de séquence système, nous procéderons à dégager le fonctionnement de création d'une offre de garde sous ses différents composants.

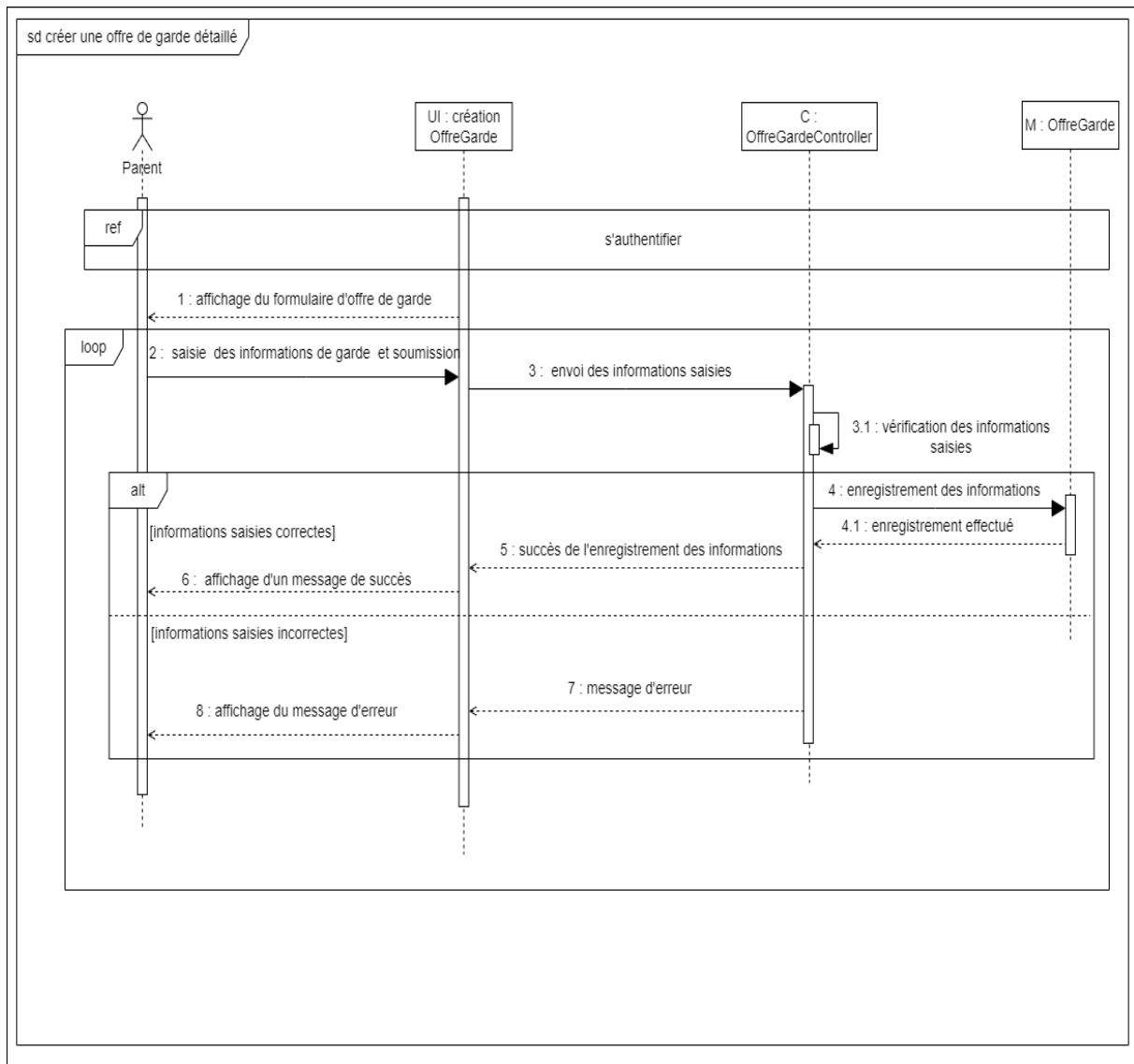


FIG. 6.6 : Diagramme de séquence détaillé du cas d'utilisation «créer une offre de garde».

### Diagramme de séquence détaillé de candidature à une offre de garde

En supplément de ce a qui a été fait dans la figure 6.4 nous ajouterons l'interaction interne de ses composants selon la figure 6.7.

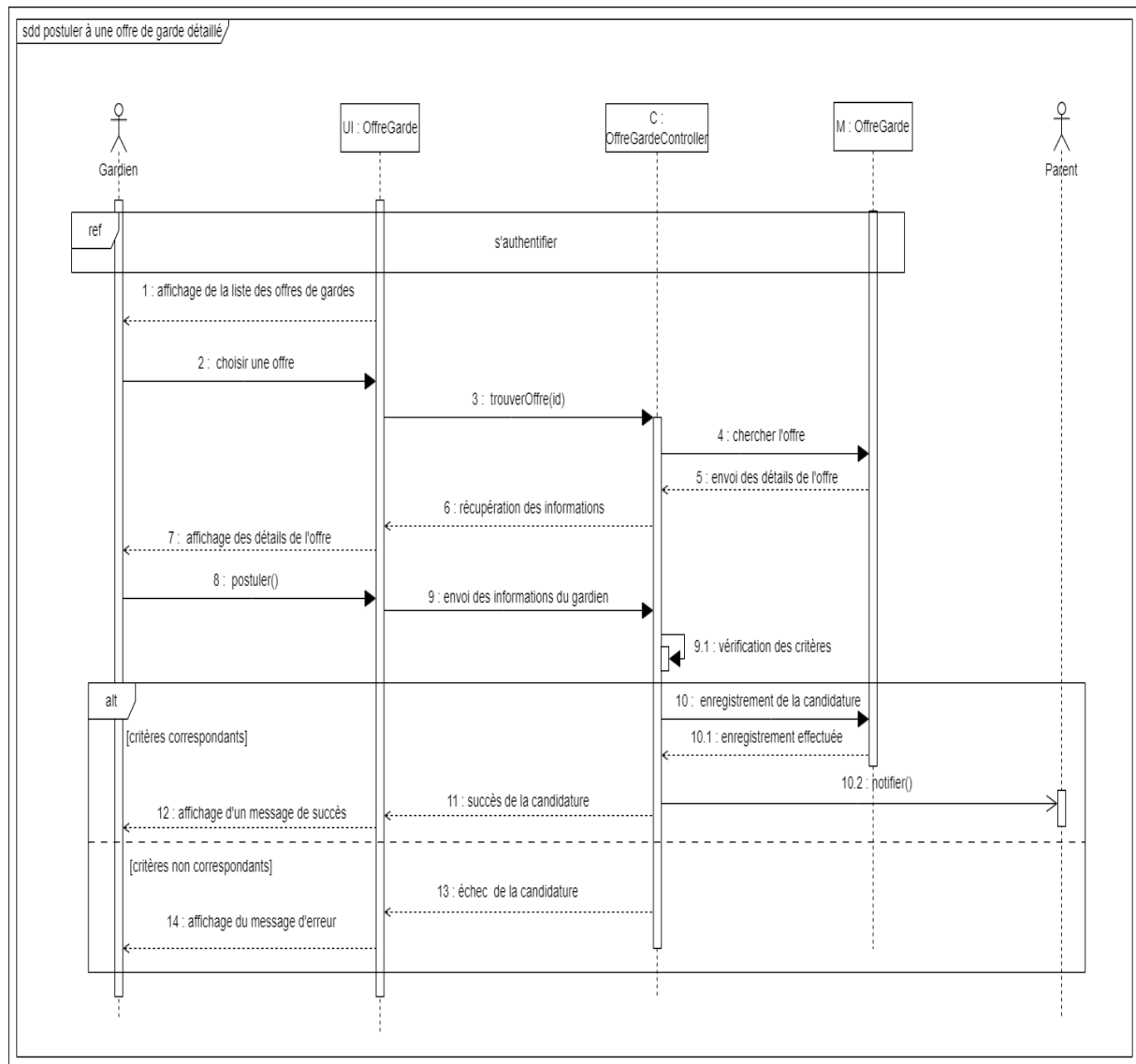


FIG. 6.7 : Diagramme de séquence détaillé du cas d'utilisation «postuler à une offre de garde».

### Diagramme de séquence détaillé de gestion d'un entretien

Afin de mieux expliquer le bon déroulement de la gestion d'un rendez vous d'entretien il était nécessaire de développer l'enchaînement de ce cas d'utilisation sous les différents composants voir figure 6.8.

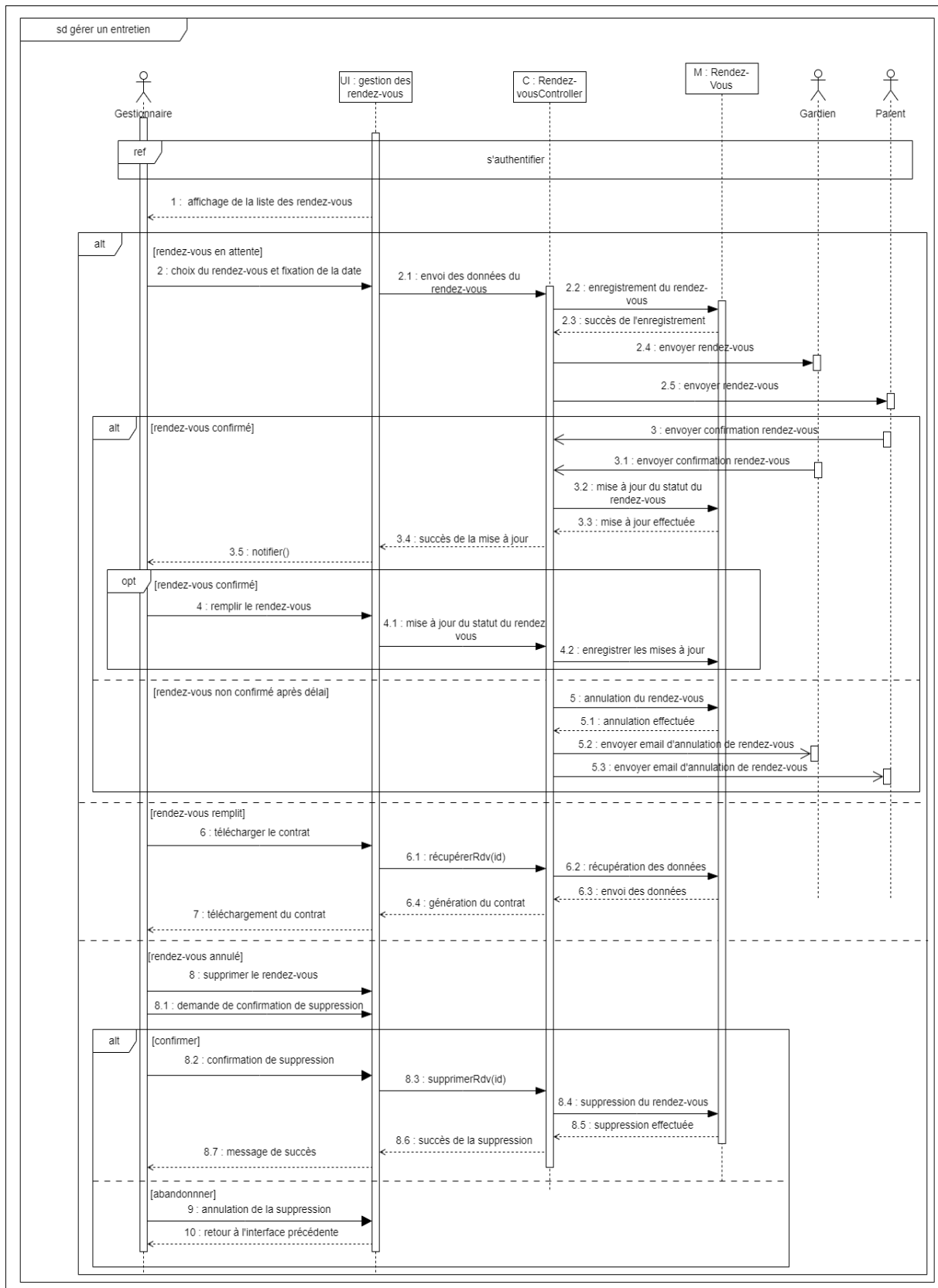


FIG. 6.8 : Diagramme de séquence détaillé du cas d'utilisation «Gérer un entretien».

### 6.5.2 Diagrammes d'activité

Afin de mieux comprendre l'enchaînement des cas d'utilisations, nous avons décidé de les enrichir des diagrammes d'activités.

#### Diagramme d'activité de création d'une offre de garde

Nous retraçons ici l'enchaînement nominal du processus de création d'une offre de garde selon la figure 6.9.

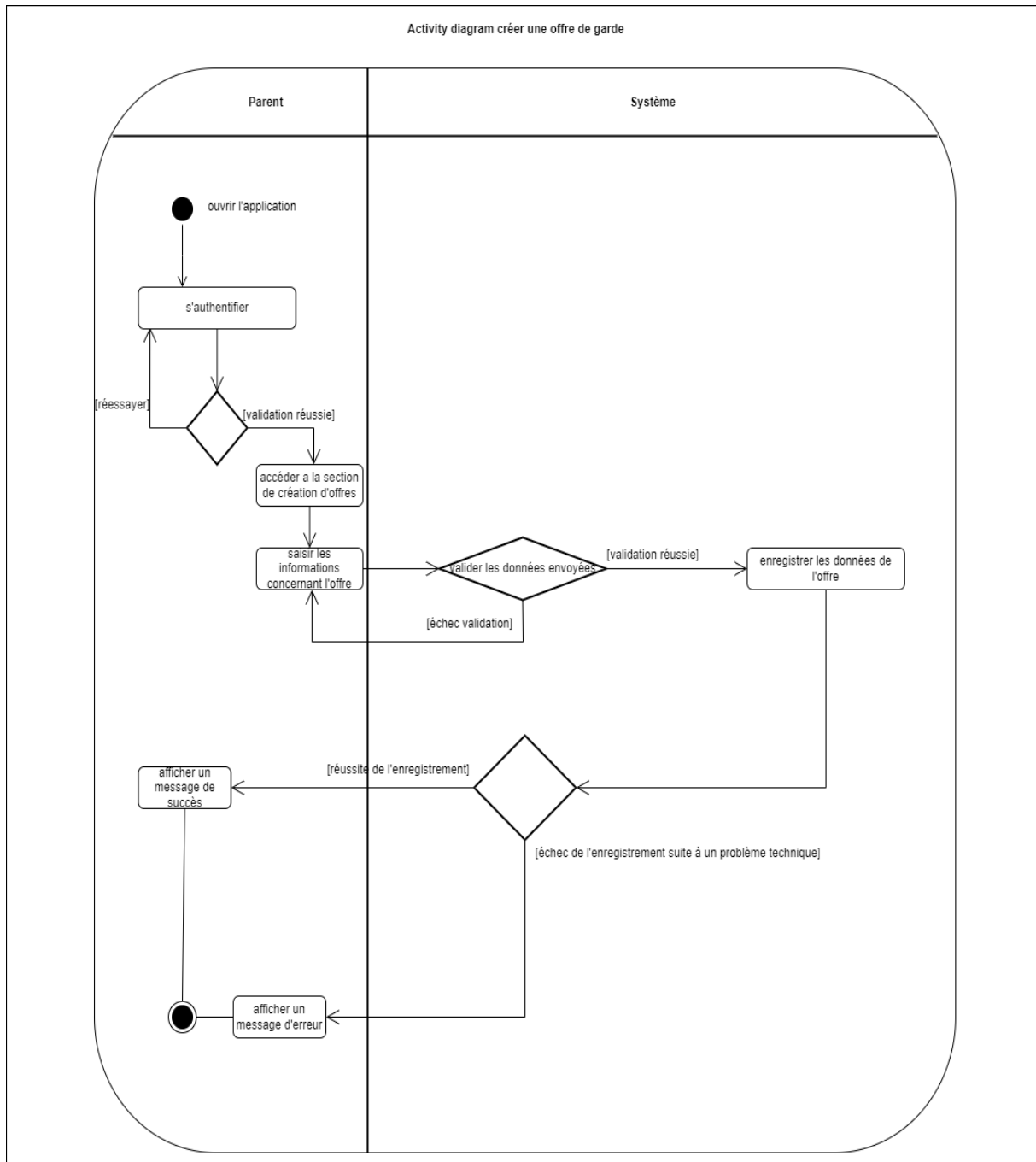


FIG. 6.9 : Diagramme d'activité du cas d'utilisation «créer une offre de garde».

### Diagramme d'activité de gestion d'un rendez-vous

Ici nous rapportons le déroulement de gestion d'un rendez-vous d'entretien entre un parent et une nourrice suite a l'approbation de sa candidature faite auprès du parent qui a créer l'offre de garde. Voir figure 6.10.

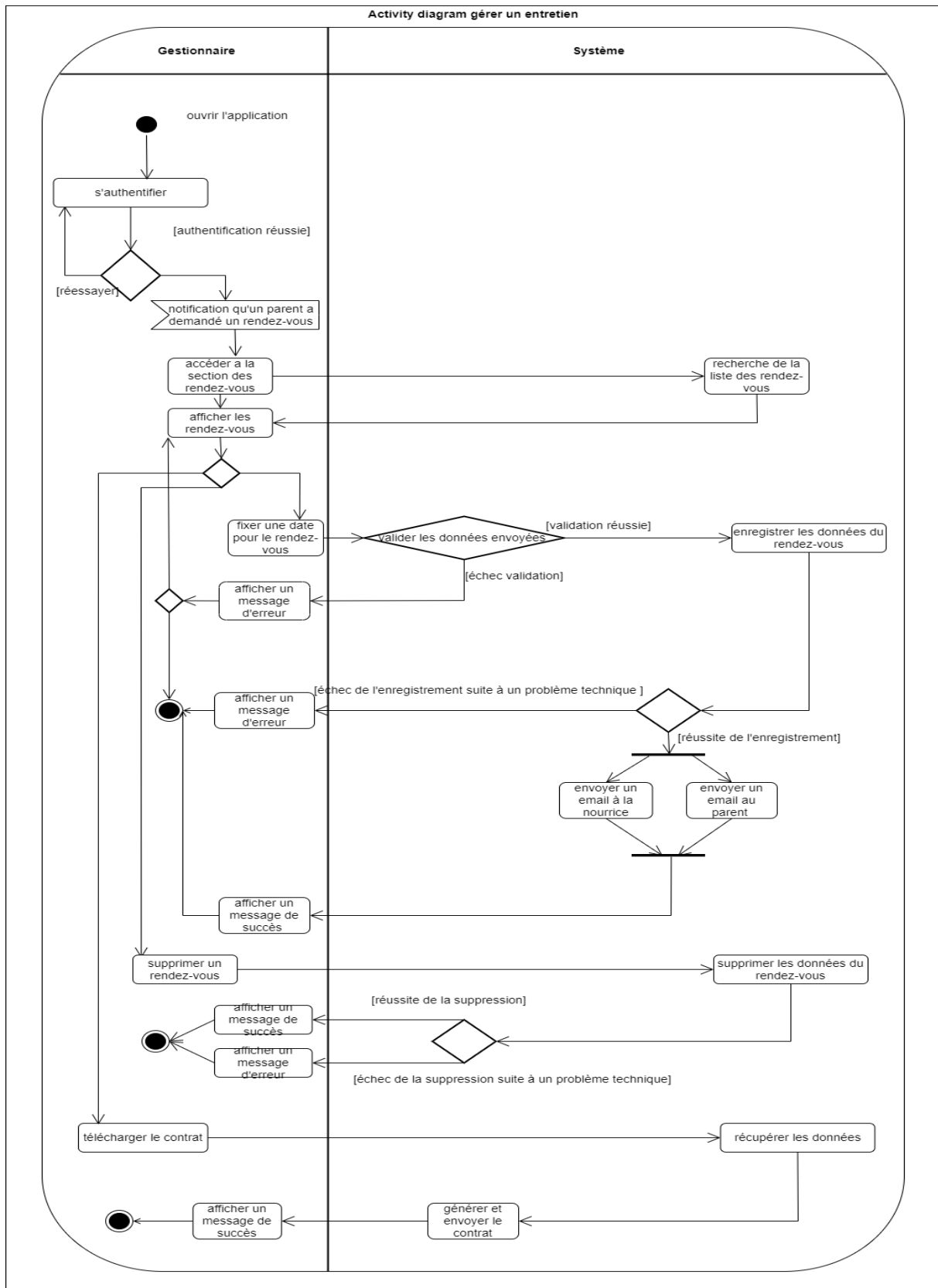


FIG. 6.10 : Diagramme d'activité du cas d'utilisation «gérer un rendez-vous».



### 6.5.3 Diagramme de classe de conception

La figure 6.11 représente le diagramme de classe conception du sprint 3, où on peut distinguer l'ajout des classes intervenant dans les fonctionnalités supplémentaires telles que la gestion des rendez-vous et les offres de garde.

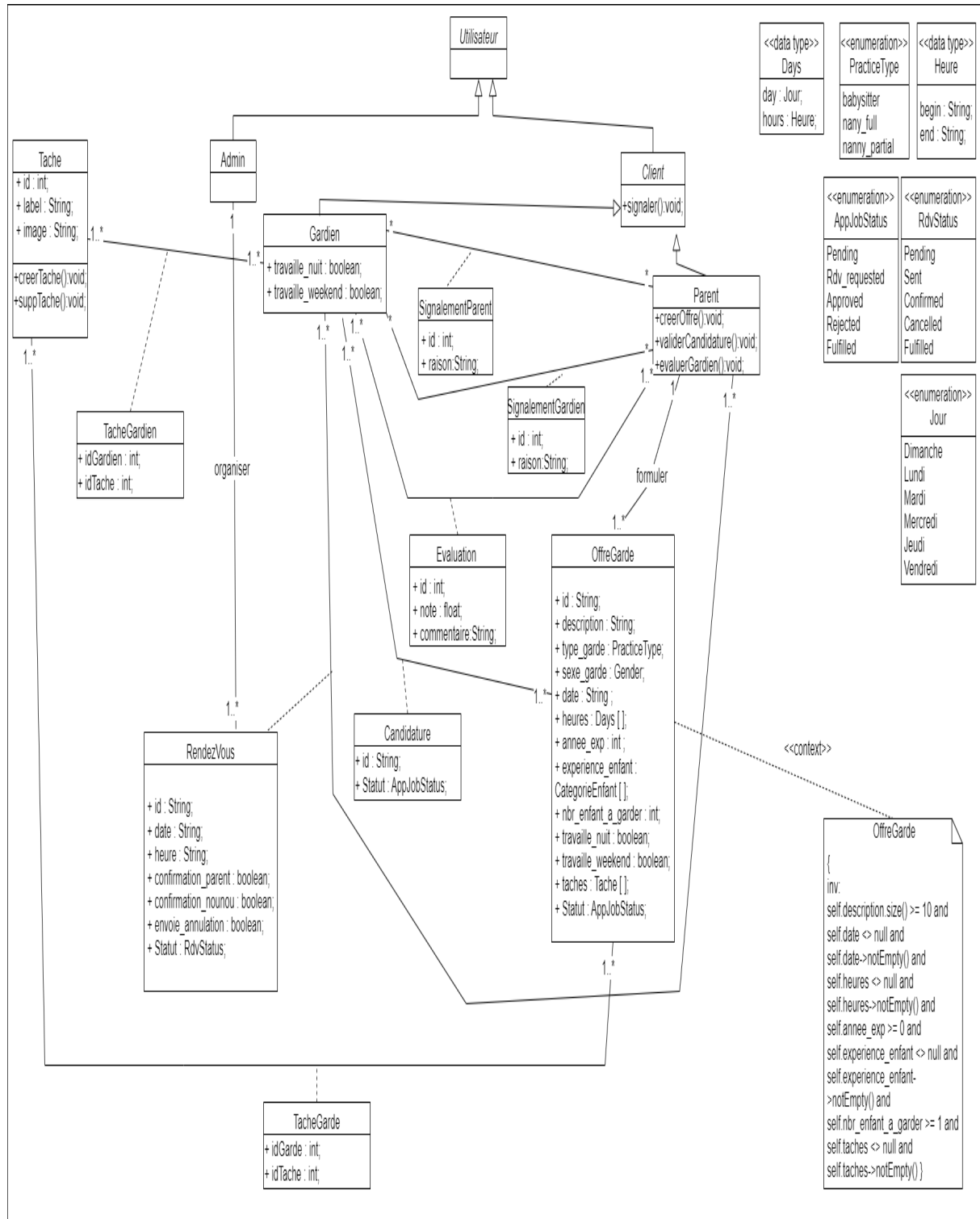


FIG. 6.11 : Diagramme de classes conception du sprint 3.

### 6.5.4 Diagramme de classes global

diagramme de classe englobe toutes les entités intervenantes dans les fonctionnalités fournies par le système. Étant donné que précédemment nous avons abordé les diagrammes de classe de chaque sprint, nous nous sommes permises de réduire quelques aspects du diagramme par souci d'organisation. Voir figure 6.12.

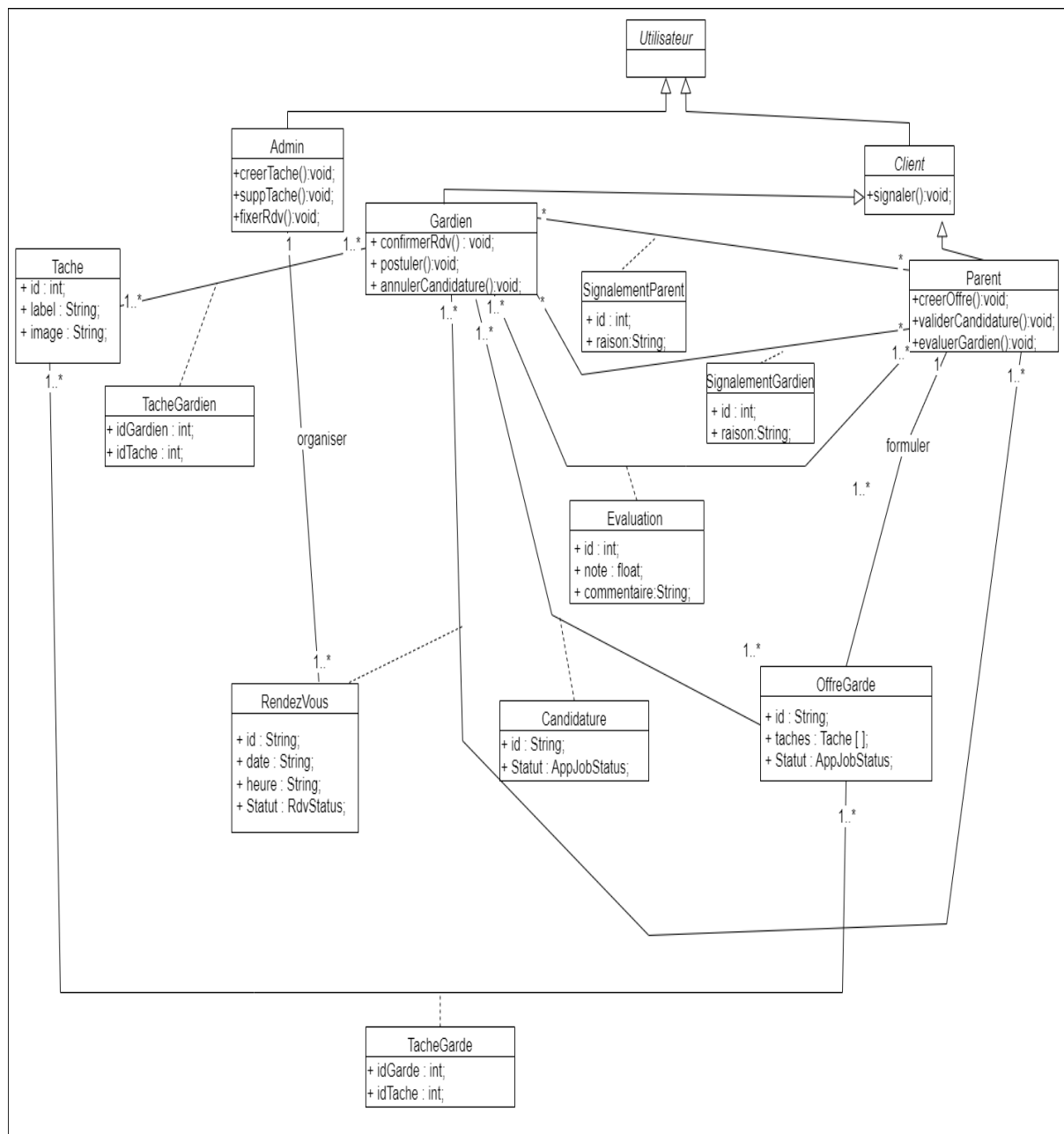


FIG. 6.12 : Diagramme des classes global.

## 6.6 Implémentation

Pendant la phase d'implémentation, nous nous attellerons à suivre la programmation afférente aux spécifications et à la conception détaillée faite auparavant, afin de mettre en oeuvre les fonctionnalités du système. Pour cela nous procéderons comme suit :

### 6.6.1 Modèle relationnel de données

En ajoutant les nouvelles tables à celles déjà définies (Admin, Parent et Gardien) dans les sprints précédents. Nous relevons que les attributs portants des (\*) sont des clefs étrangères venant d'autres tables et on obtient alors le modèle relationnel suivant :

Admin(id, idRendezVous\*, pseudo, email, password, roles, privileges).

Parent(id, idOffreGarde\*, email, nom, prenom, password, sexe, etat-civile, telephone, codePostale, adresse, nbr-enfant, motif-bannissement, statut).

Gardien(id, email, nom, prenom, password, sexe, type-garde, etat-civile, telephone, codePostale, adresse, nbr-enfant-a-garder, annee-exp, experience-enfant, tarif-par-enfant, evaluation, biography, cv, image, motif-bannissement, travaille-nuit, travaille-weekend, statut).

OffreGarde(id, description, type-garde, sexe-garde, date, heures, nbr-enfant-a-garder, annee-exp, experience-enfant, taches, travaille-nuit, travaille-weekend, statut).

Candidature(id, idOffreGarde\*, idGardien\*, statut).

Evaluation(id, idParent\*, idGardien\*, note, commentaire).

SignalementGardien(id, idGardien\*, idParent\*, raison).

SignalementParent(id, idParent\*, idGardien\*, raison).

RendezVous(id, idParent\*, idGardien\*, date, heure, confirmationParent, confirmation-Gardien, envoi-annulation, statut).

Tache(id, label, image).

TacheGarde(idTache\*, idGarde\*).

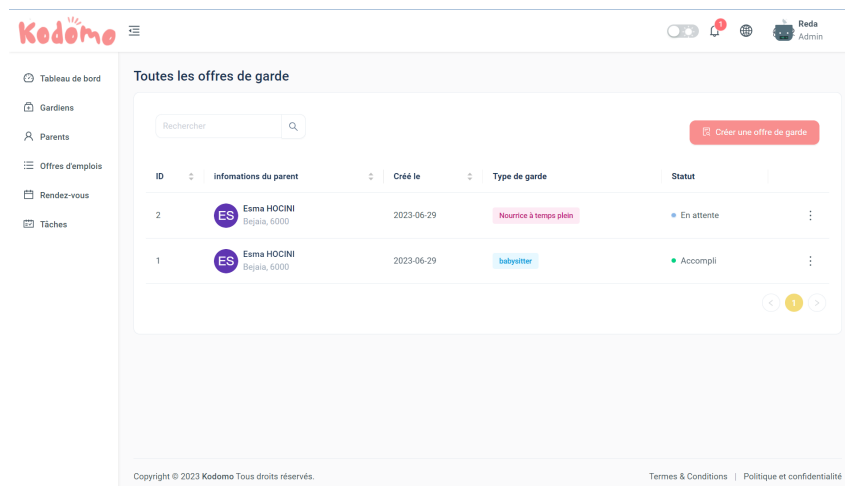
TacheGarde(idTache\*, idGardien\*).

### 6.7 Présentation d'interfaces

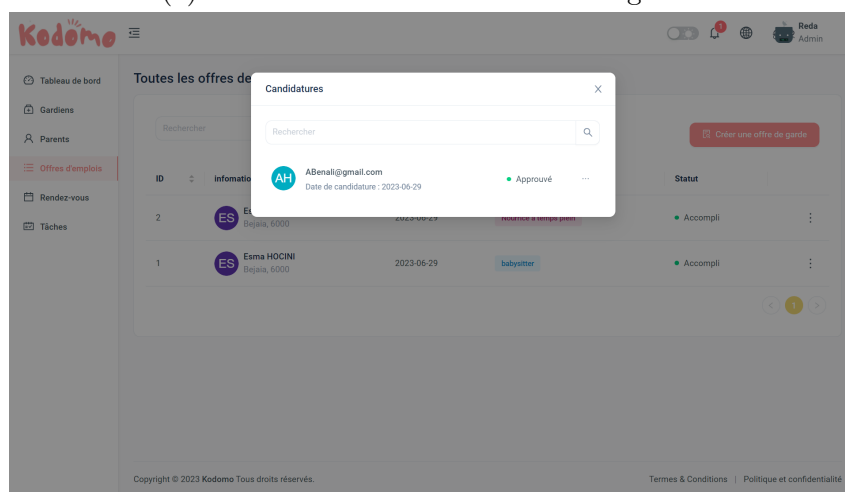
Dans cette section nous allons montrer quelques interfaces de notre plate-forme "KODOMO" relatives au Sprint 3.

#### 6.7.1 Interface de gestion des offres de gardes

C'est ici qu'on peut consulter et gérer la totalité des offres de garde effectuées par les parents dans l'application. On peut supprimer une offre, la modifier, ou même consulter la liste des gardiens ayant postulé à cette dernière. On peut aussi postuler à cette offre pour un gardien désigné. Le bouton "Ajouter une offre d'emploi" nous permet d'ajouter une nouvelle offre de garde pour un parent donné. Voir figures 6.13 et 6.14.

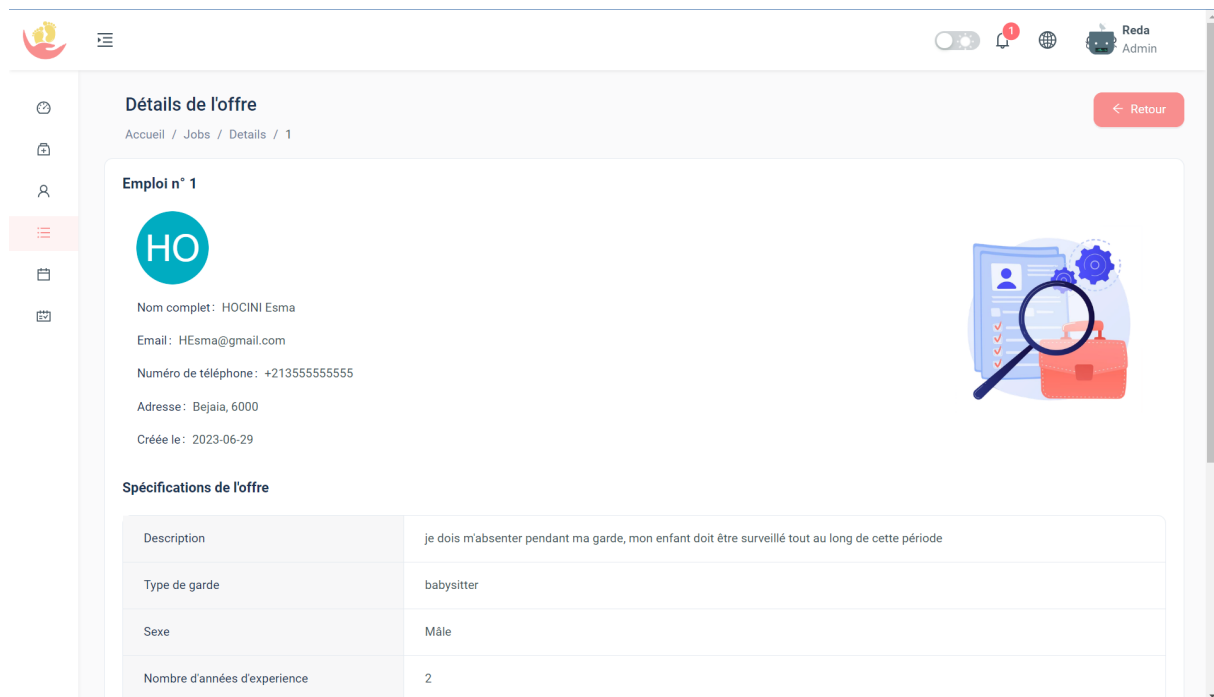


(a) Interface de la liste des offres de garde.

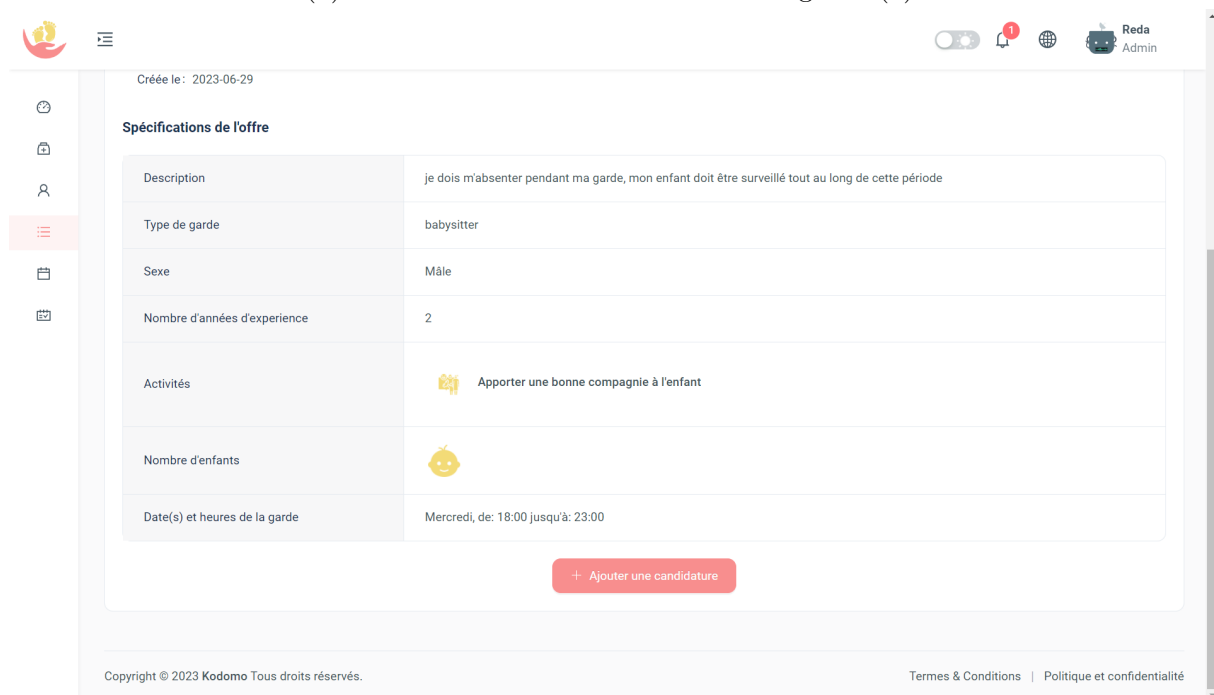


(b) Liste des candidatures d'une offre de garde.

FIG. 6.13 : Interface de gestion des offres de garde.



(a) Interface des détails d'une offre de garde (1).

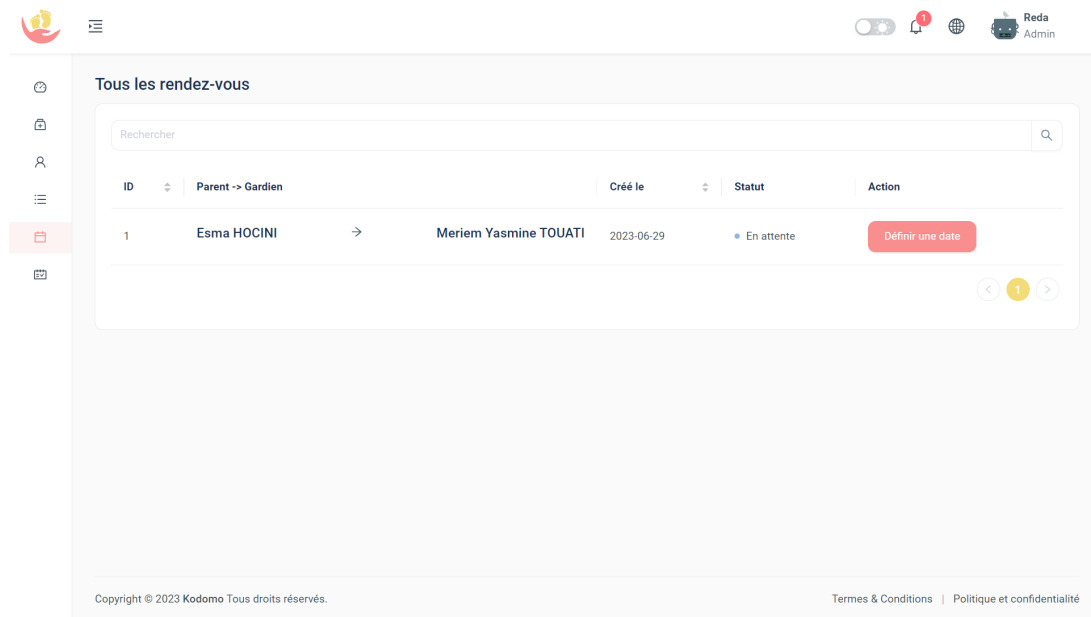


(b) Interface des détails d'une offre de garde (2).

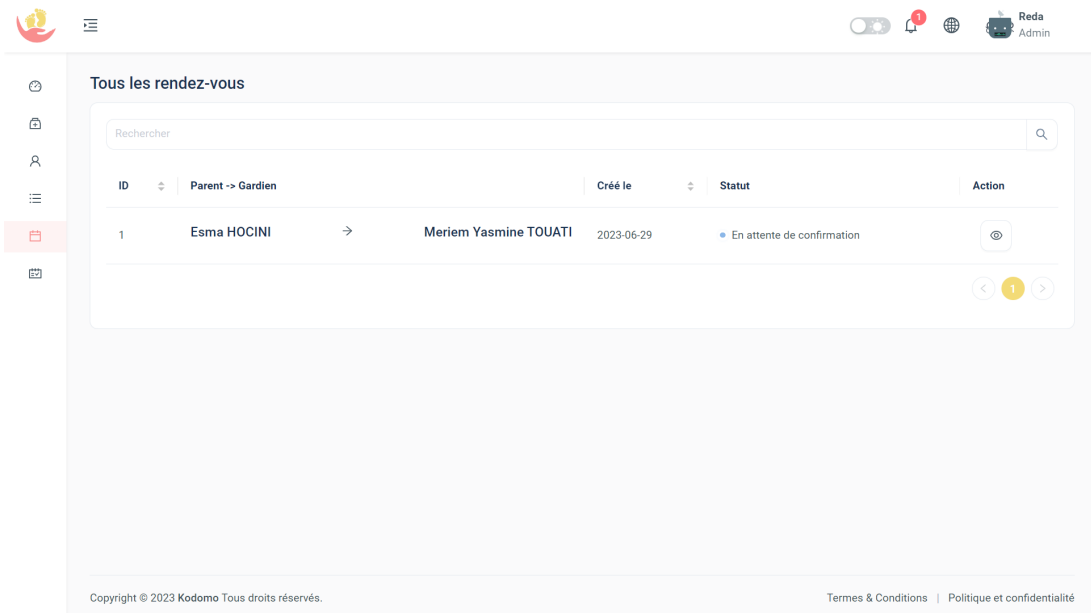
FIG. 6.14 : Interface de gestion des offres de garde.(suite)

### Interface de gestion des rendez-vous d'entretiens

Quand un parent, envoie une demande d'entretien avec le gardien qu'il a choisi, sa demande s'enregistre dans cette section. Au début le statut de la demande est en attente, jusqu'à ce que l'administrateur ou le gestionnaire fixe de son côté une date de son choix, en cliquant sur le bouton "Fixer une date". Une fois que l'entretien est achevé et le contrat signé entre les parties prenantes, il peut revenir dans cette section pour changer le statut à "effectué" et puis générer le contrat de garde numérique. Voir figure 6.15.



(a) Interface de la liste des rendez-vous.



(b) Demande de rendez-vous suite à une validation de candidature.

FIG. 6.15 : Liste des rendez-vous

### 6.8 Conclusion

Le sprint 3 a joué un rôle essentiel dans le développement de notre projet, en nous permettant de consolider et d'enrichir notre plate-forme. Grâce aux efforts déployés pendant ce sprint, nous avons atteint un jalon important : la livraison d'une version bêta de notre travail. Il convient néanmoins de souligner que ce projet se concentre principalement sur l'implémentation des fonctionnalités de base. De ce fait, plusieurs améliorations et fonctionnalités pourront l'enrichir et être intégrées ultérieurement.

# Conclusion et perspectives

Dans ce mémoire, nous avons exploré le domaine de la garde d'enfants et avons identifié les défis auxquels les parents sont confrontés lorsqu'ils recherchent des services de garde fiables et adaptés. Nous avons proposé la création d'une plate-forme dédiée, baptisée "KODOMO", qui offre une solution pratique et sécurisée pour mettre en relation les parents et les gardiens qualifiés.

L'objectif principal de ce mémoire était de concevoir et de mettre en œuvre cette plate-forme afin de répondre aux besoins réels des parents et de faciliter leur recherche de services de garde adaptés à leurs enfants. Nous avons utilisé des méthodologies de développement telles que Scrum et UP, ainsi que des outils comme UML et OCL, pour guider notre processus de développement.

Au cours de notre étude, nous avons réalisé une analyse approfondie du domaine métier, identifié les forces et les faiblesses de notre solution, et mené une enquête auprès des crèches locales pour consolider nos résultats.

Dans le cadre de notre processus de développement, nous avons adopté une approche agile qui nous a permis de livrer régulièrement des fonctionnalités et de répondre de manière flexible aux besoins des utilisateurs. Cette approche a présenté de nombreux avantages significatifs, tels qu'une meilleure visibilité sur l'avancement du projet, une capacité à s'adapter aux changements et une gestion améliorée des risques. Cependant, nous avons également identifié certaines limites, notamment la nécessité d'une coordination et d'une communication solides au sein de l'équipe.

En conclusion, ce travail nous a permis de développer des compétences pendant notre stage, et d'étendre les connaissances pratiques que nous avons acquises dans le domaine du développement logiciel. Avec l'espoir d'avoir contribué humblement à solutionner la problématique de garde d'enfants dans notre société.

En ce qui concerne les perspectives d'avenir, plusieurs pistes intéressantes se présentent pour améliorer les services proposés par KODOMO et offrir une meilleure expérience aux parents. Voici quelques perspectives à considérer :

- Développer une application mobile dédiée pour les parents et les gardiens.
- Introduire des abonnements payants donnant accès à des services supplémentaires aux parents.
- Collaborer avec des garderies de la petite enfance pour élargir l'offre de services.
- Ajouter la possibilité de paiement en ligne pour plus de commodité et de sécurité.



# Bibliographie

- [1] *Crèche ou nounou : le casse-tête des parents en cette nouvelle rentrée.* <https://www.tsa-algerie.dz/creche-ou-nounou-le-casse-tete-des-parents-en-cette-nouvelle-rentree/>. Paru le 08 Sept. 2018, Consulté le 02 mai. 2023.
- [2] *Crèches : normes et réglementation.* <https://www.algerie360.com/creches-normes-et-reglementation/>. Paru le 2 Oct. 2012, Consulté le 02 mai. 2023.
- [3] *Crèches et garderies : le grand calvaire des parents.* <https://www.jeune-independent.net/creches-et-garderies-le-grand-calvaire-des-parents/>. Paru le 01 Oct. 2022, Consulté le 03 mai 2023.
- [4] Martin FOWLER. *A Brief Guide to the Standard Object Modeling Language.* 2003.
- [5] *French Babysitter Algeria.* <https://www.jobandsalaryabroad.com/fr/algeria/french-babysitter-algeria.html>. Consulté le 03 mai 2023.
- [6] JOBTED. *Fiche métier : Baby-sitter.* <https://fr.jobted.com/fiche-m%C3%A9tier/baby-sitter>. 2023.
- [7] *Journal Officiel de la République Algérienne Démocratique et Populaire.* Conventions et Accords Internationaux - Lois et Décrets, Arrêtés, Décisions, Avis, Communications et Annonces (Traduction française). Consulté le 03 mai 2023.
- [8] *Le projet des crèches municipales.* <https://www.vitaminedz.com/fr/Algerie/le-projet-des-creches-municipales-780815-Articles-0-18300-1.html>. Paru le 20 Sept. 2011, Consulté le 03 mai 2023.
- [9] *Méthode agile Scrum : gestion de projet.* <https://blog.trello.com/fr/methode-agile-scrum-gestion-projet/>. Paru le 05 Juin. 2017, Consulté le 25 mai 2023.
- [10] *Méthode Scrum : les bénéfices pour les développements web.* <https://www.bocasay.com/fr/methode-scrum-benefices-developpements-web/>. Paru le 25 Mars. 2022, Consulté le 25 mai 2023.
- [11] OFFICE NATIONAL DES STATISTIQUES (ONS). *Démographie Algérienne 2020.* <https://www.ons.dz/IMG/pdf/Demographie%20Algérienne2020.pdf>. Paru en 2020, Consulté le 10 avril 2023.
- [12] *Places maximum pour une assistante maternelle.* <https://www.nounou-top.fr/info/places-maximum-assmat>. Consulté le 20 mai 2023.
- [13] *Population.* <http://www.dsp-bejaia.dz/index.php/population>. Consulté le 02 mai 2023.
- [14] Roger S. PRESSMAN. *Software Engineering : A Practitioner's Approach.* 7th edition, 2019, p. 44.

- [15] Ken SCHWABER. *Agile Software Development with Scrum*. Upper Saddle River, NJ : Prentice Hall, 2004, p. 32-34.
- [16] *Scrum, c'est quoi ? Définition de Scrum*. <https://www.qrpinternational.fr/blog/glossaire/scrum-cest-quoi-definition-scrum/>. Paru le 12 Janvier. 2021, Consulté le 25 mai 2023.
- [17] P.-A. SUNIER. "Le Processus unifié de développement logiciel". In : Cours "Méthodologies de développement de logiciels de gestion", Haute Ecole Arc Neuchâtel Berne Jura. 2005. Chap. 6.
- [18] *The Astrology Page - Unified Process*. En ligne. Paru en 2023, Consulté le 20 avril 2023. URL : <https://fr.theastrologypage.com/unified-process>.
- [19] *Titre de la page*. En ligne. Consulté le 20 avril 2023. URL : <http://www.xr6805.fr/IS/Diagrammedecontexte.html>.
- [20] *UML - Définitions*. <https://www.futura-sciences.com/tech/definitions/informatique-uml-3979/>. Paru le 22 Aout. 2021, Consulté le 25 mai 2023.

# Annexes

# Annexe A

## Quelques technologies utilisées

### A.1 Javascript

JavaScript est un langage de programmation mettant en œuvre des fonctionnalités complexes sur des pages web. Principalement utilisé dans le navigateur web, permettant aux développeurs de manipuler le contenu des pages web via le DOM, de manipuler des données avec AJAX et IndexedDB, de dessiner des graphiques avec canvas, d'interagir avec le dispositif exécutant le navigateur via diverses API, à noter qu'une page web ne se contente pas d'un simple affichage d'informations statiques, qu'il s'agisse de mises à jour de contenu, de cartes interactives, de graphiques 2D/3D animés, de juke-boxes vidéo défilants ou autres, JavaScript est forcément impliqué.

### A.2 Typescript

TypeScript est un langage de programmation open source développé par Microsoft. Il s'agit d'un sur-ensemble typé de JavaScript qui est compilé en JavaScript standard et peut être utilisé pour le développement d'applications côté client et serveur. TypeScript offre une syntaxe plus concise et une meilleure gestion des erreurs de syntaxe et d'exécution que JavaScript. Il prend également en charge les fonctionnalités de programmation orientée objet telles que les classes, les interfaces, les types, les décorateurs et les modules. TypeScript est conçu pour améliorer la qualité, la maintenabilité et la lisibilité du code, ainsi que pour améliorer la productivité des développeurs grâce à son support pour la saisie automatique et la documentation du code.

### A.3 MySQL

MySQL est un système de gestion de base de données relationnelle (SGBDR) populaire et largement utilisé. Il est open source, ce qui signifie que son code source est disponible gratuitement et peut être modifié et distribué.

### A.4 Node js

Node.js est un environnement d'exécution open-source côté serveur qui permet d'exécuter du code JavaScript en dehors d'un navigateur web. Il est construit sur le moteur JavaScript V8 de Google Chrome et fournit des fonctionnalités telles que les entrées/sorties réseau, la gestion des fichiers, la communication en temps réel et l'exécution de requêtes HTTP. Node.js est utilisé pour construire des applications web, des outils en ligne de commande, des serveurs et des applications de type backend pour les applications mobiles ou desktop. Il est populaire pour sa rapidité d'exécution, sa scalabilité, sa facilité de mise en place et sa grande communauté de développeurs 8.

### A.5 Npm

NPM (Node Package Manager) est un gestionnaire de packages pour l'écosystème Node.js. Il permet aux développeurs de partager et de réutiliser du code facilement, ainsi que de gérer les dépendances des projets Node.js. Il permet également d'installer des packages tiers, des outils de développement et des bibliothèques pour les projets Node.js.

#### A.5.1 REST

Une API REST (Representational State Transfer) est un style d'architecture logicielle pour la conception de services web. Elle est basée sur les principes fondamentaux du protocole HTTP, tels que les méthodes de requête (GET, POST, PUT, DELETE) et les codes de statut de réponse (200 OK, 404 Not Found, etc.).

#### A.5.2 HTTP

HTTP (Hypertext Transfer Protocol) est un protocole de communication utilisé pour le transfert de données sur le Web. Il définit la structure et les règles d'échange d'informations entre un client (généralement un navigateur web) et un serveur web.

### A.6 WebSocket

WebSocket est un protocole de communication bidirectionnelle et en temps réel, qui permet une communication interactive entre un navigateur web (client) et un serveur. Contrairement au protocole HTTP traditionnel qui suit un modèle de requête-réponse, WebSocket permet une connexion persistante entre le client et le serveur, permettant ainsi l'échange de données en temps réel dans les deux sens.

## A.7 Bcrypt

Bcrypt est une fonction de hachage de mots de passe sécurisée. Il s'agit d'un algorithme de hachage basé sur le chiffrement de "Blowfish". L'objectif principal de Bcrypt est de protéger les mots de passe en les rendant difficiles à décrypter, même si la base de données qui les stocke est compromise.

# Annexe B

## Diagrammes de séquence système

Par manque d'espace, nous n'avons pu intégrer ces diagrammes dans le chapitre 4, et donc les voici :

### B.1 Diagramme de séquence système de création de compte

Aussi trivial qu'important dans toute application informatique, au même titre que l'authentification, le diagramme de séquence système du cas d'utilisation «créer un compte» décrit l'enchaînement basique de saisie d'email et de mot de passe.

### B.2 Diagramme de séquence système de réinitialisation de mot de passe

De même pour la création, la réinitialisation de mot de passe est une fonctionnalité incontournable dans une application informatique. Le diagramme du cas d'utilisation «réinitialisation de mot de passe» retrace la procédure de déroulement de cette dernière.

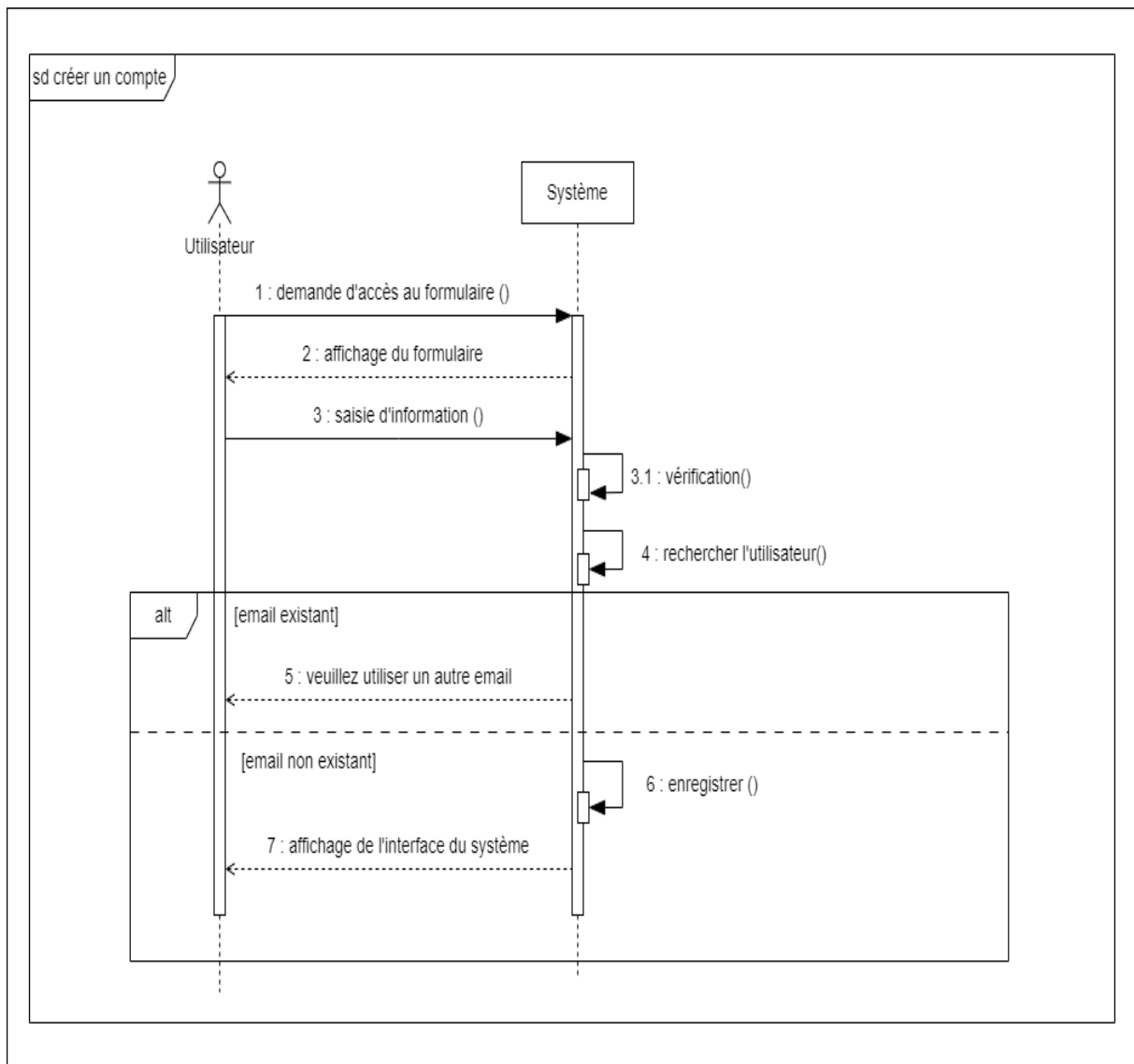


FIG. B.1 : Diagramme de séquence système du cas d'utilisation «créer un compte».



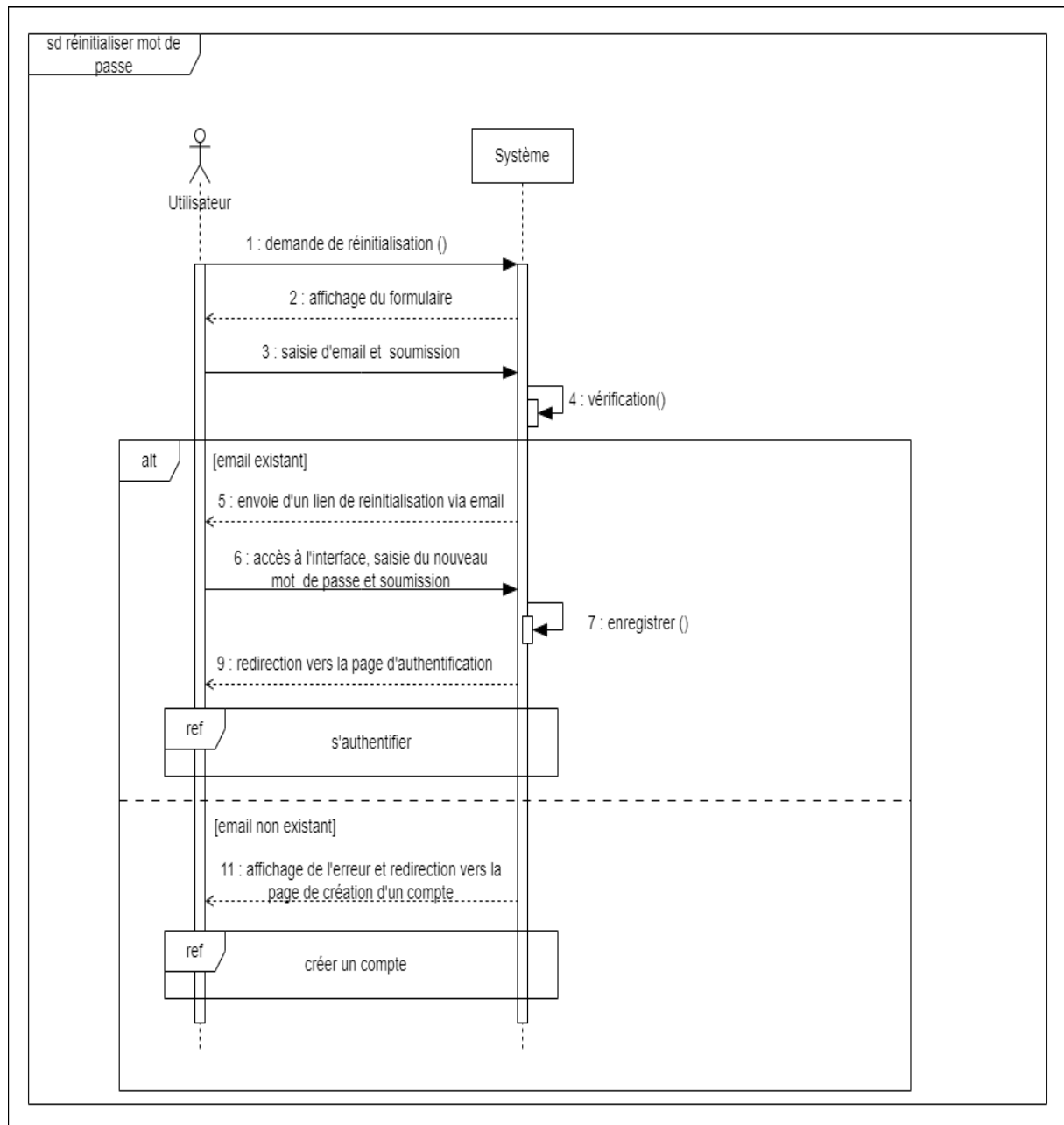


FIG. B.2 : Diagramme de séquence système du cas d'utilisation «réinitialisation de mot de passe».

# Annexe C

## Diagramme de Gantt des Sprints

Tâches	Sprint 1	Sprint 2	Sprint 3
Configuration et installation dépendances	■		
Mise en oeuvre du manuel d'installation de la partie Frontend et Backend	■		
Création de la BDD	■		
Configuration des routes publiques et privées	■		
Configartion du thème mode jour et mode nuit	■		
Configuration du système de traduction du site	■		
Interface "Connexion"/"Inscription"	■		
Liaison Backend des fonctionnalités d'authentification, d'inscription et du mot de passe oublié	■		
Test de fonctionnement	■		
Interface de la liste des gardiens, ajout, modification, suppression, et affichage du profil d'un seul gardien, formulaire de blocage	■		
Liaison Backend pour la gestion des gardiens	■		
Test de fonctionnement	■		
Interface de la liste des parents, ajout, modification, suppression, et affichage du profil d'un seul parent, formulaire de blocage	■		
Liaison Backend pour la gestion des parents	■		
Test de fonctionnement	■		
Interface des gestionnaires (liste, ajout, modification des privilèges)		■	
Liaison Backend pour la gestion des gestionnaires		■	
Test de fonctionnement		■	
Interface des Dashboards (Tableaux de bord) pour les statistiques du système		■	
Liaison Backend avec la possibilité d'imprimer les résultats		■	
Test de fonctionnement		■	
Interface de la liste des offres de garde, ajout, modification, suppression et affichage d'une seule offre			■
Liaison Backend pour la gestion des offres de garde			■
Test de fonctionnement			■
Liaison Backend pour la gestion des candidatures des gardiens			■
Tests et documentation d'API			■
Interface de la liste des demandes d'entretiens, organisation d'un entretien			■
Liaison Backend pour la gestion des candidatures des entretiens avec la génération d'un contrat de garde numérique			■
Test de fonctionnement			■
Interface de la liste des tâches, ajout et suppression			■
Liaison Backend pour la gestion des taches			■
Test de fonctionnement			■
Ajout des fonctionnalités de filtrage, de recherche et de tri des différents résultats			■
Ajout d'un système de notifications			■
Génération de la documentation d'API			■

TAB. C.1 : Diagramme de Gantt des Sprints.

## Annexe D

### Descriptifs des réunions effectuées

Id réunion	Date	Objet	Description
RN-01	16/11/2022	Introduction	Sollicitation de l'entreprise pour une demande de stage, discussion du processus de travail, de la date du début de stage, ainsi que les thèmes que l'entreprise propose.
RN-02	18/11/2022	Introduction	Communication du choix définitif du thème
RN-03	25/12/2022	Brainstorming Design	Discussion du nom de l'application et mise en œuvre de la charte graphique.
RN-04	03/01/2023	Brainstorming	Brève étude de l'existant du domaine métier.
RN-05	27/02/2023	Analyse	Collecte des besoins fonctionnels auprès du Product Owner (Première version du Product Backlog).
RN-06	03/03/2023	Analyse	Mise à jour du Product Backlog.
RN-07	18/03/2023	Planification	Répartition des tâches pour le premier Sprint.
RN-08	08/04/2023	Vérification	Vérification du Scrum Master de notre état d'avancement dans les tâches du Sprint courant.

Id réunion	Date	Objet	Description
RN-09	11/04/2023	Bugs	Sollicitation du Scrum Master vis-à-vis d'un bug rencontré lors du développement.
RN-10	16/04/2023	Planification	Répartition des tâches pour le deuxième Sprint et apport des corrections au Sprint précédent (Backlog de Sprint).
RN-11	27/04/2023	Vérification	Vérification du Scrum Master de notre état d'avancement dans les tâches du Sprint courant.
RN-12	05/05/2023	Planification	Répartition des tâches pour le troisième Sprint et apport des corrections au Sprint précédent (Backlog de Sprint).
RN-13	13/05/2023	Vérification	Vérification du Scrum Master de notre état d'avancement dans les tâches du Sprint courant.

TAB. D.1 : Tableau des réunions.

# Annexe E

## Questionnaire pour les crèches

1. Combien d'enfants sont ils admis au sein de votre établissement ?
2. Quel est l'intervalle d'âges de ces derniers ?
3. Comment est ce que vous segmentez et intervalle pour prendre en charge les enfants ?
4. Comment gérez vous les différentes attentes des enfants en terme de leur divertissement ou bien centre d'intérêts ?
5. Quels sont vos horaires d'ouverture / fermeture ?
6. Sur quelle fréquence vous arrive t-il d'étendre vos horaires en raisons du retards des parents, cela vous dérange t-il ?
7. Vous arrive t-il d'être dépassé par la charge de travail ?
8. Proposez vous un service ou une nourrice s'engagerait à garder un enfant à domicile ?
9. Quels sont vos jours de travail ?
10. Quelles sont vos tarifications ?
11. Proposez vous une formule, si oui précisez la quelle ?
12. Comment communiquez vous avec les parents dans les différents cas (informationnel, urgence, etc.) ?
13. Par quels moyens les parents parviennent-ils à trouver votre établissement ?