

République Algérienne Démocratique et Populaire

**Ministère de l'Enseignement Supérieur et de la Recherche
Scientifique**

Université Abderrahmane Mira - Béjaïa -

Département d'Informatique



Mémoire de fin de cycle master

En vue de l'obtention du diplôme de Master Professionnel en Informatique

Option : Génie Logiciel

**Développement d'une application web dynamique dédiée
aux animaux domestiques**

Réalisé par :

Mlle. Laidi Melissa & Mlle.Sellami Asma

Soutenu le 02 juillet 2024, devant le jury composé de :

Président :Mr. ATMANI Mouloud (MCA)

Encadrant :Mr. ZERARGA Loutfi (MCB)

Examinatrice :Mme. BELKHIRI Louiza (MCB)

Promotion : 2023/2024

Dédicace

"À mes compagnons à quatre pattes, Tsuki et Picsou, dont la présence inspirante a allumé la flamme de la création. Vos câlins réconfortants et vos moments ludiques m'ont guidé vers le chemin de l'innovation. Grâce à vous, cette application dédiée au bien-être de tous nos précieux animaux de compagnie a vu le jour. Portée par l'étreinte de votre amour, l'idée a germé, faisant fleurir ce projet dédié à vous, mes fidèles compagnons."

Melissa

Dédicace

"Avec tous mes sincères sentiments, je dédie mon projet à mon paradis, mon premier amour, ma lumière dans l'obscurité, ma chère maman qui s'est sacrifiée pour que je puisse étudier et apprendre et pour me sortir des ténèbres de l'ignorance pour m'amener vers la lumière de la connaissance depuis mon enfance jusqu'à aujourd'hui. Sans oublier mon cher papa, ainsi que tous les membres de ma famille et notre cher encadrant qui nous a encouragés à accomplir ce projet à terme."

Asma

Remerciement

« Au début de ce travail, nous souhaitons exprimer notre profonde gratitude envers Dieu le Tout Puissant pour nous avoir accordé le courage et la patience nécessaires pour mener ce projet à son terme.

Nous tenons également à remercier sincèrement tous les membres du jury pour avoir accepté d'examiner notre travail. Un grand merci également à nos familles pour leur soutien inconditionnel, leurs conseils précieux et leur encouragement constant.

Nos remerciements les plus sincères vont à notre encadrant, Mr. Zerarga Loutfi, pour sa précieuse guidance tout au long de ce travail, sa disponibilité et son dévouement à la réussite de ce projet.

Enfin, nous tenons à remercier tous ceux qui ont contribué à la réalisation de ce travail.»

Table des matières

Table des matières	IV
Table des figures	VIII
Liste des tableaux	X
1 Contexte d'étude et recueil des besoins	2
1.1 Introduction	2
1.2 Concepts fondamentaux du web	2
1.2.1 Fondamentaux des Technologies Web	2
1.2.2 Concepts fondamentaux du web	2
1.2.3 Classification des sites web	2
1.2.4 Architecture à trois tiers	3
1.3 Contexte de l'adoption et du sauvetage des animaux domestiques	3
1.3.1 Problématique	3
1.3.2 Rôle de « Pettracker » dans ce contexte	4
1.3.3 Solutions proposées et mission de l'application	5
1.4 Facteurs clés du succès de l'application	5
1.5 Plateformes existantes et besoins non comblés	5
1.5.1 Étude des applications existantes	6
1.5.2 Étude comparative	7
1.5.3 Description des Fonctionnalités	8
1.5.4 Analyse comparative	8
1.6 Conclusion	8
2 Méthodologie de la conception et spécification des besoins	10
2.1 Introduction	10
2.2 Méthodologie de la conception	10
2.2.1 Qu'est-ce qu'une Méthode agile?	10
2.2.2 Méthode Scrum	11
2.2.3 Langage de modélisation UML	12
2.2.4 Modèle MVC	13
2.3 Pilotage de projet avec Scrum	14
2.3.1 Rôle et user stories	14
2.3.2 User Stories	14
2.3.3 Identification des acteurs	16
2.3.4 Diagramme de Contexte	16
2.3.5 Diagramme de Cas d'Utilisation	17
2.3.6 Planification des releases	18
2.3.7 Backlog-Product (carnet de produit)	19
2.4 Charte graphique	21

2.4.1	Logotype	21
2.4.2	Couleurs dominantes	21
2.5	Conclusion	22
3	Premier Release	23
3.1	Introduction	23
3.2	Sprint 1 : Gestion des comptes utilisateurs	23
3.2.1	Diagramme de cas d'utilisation du sprint 1	24
3.2.2	Cas d'utilisation « Inscription »	25
3.2.3	Cas d'utilisation « Se connecter »	27
3.2.4	diagramme de classe de sprint01	28
3.3	sprint 2 : Gestion des profils des animaux	29
3.3.1	Diagramme de cas d'utilisation du sprint 2	29
3.3.2	Cas d'utilisation « Créer des Profils pour les animaux domestiques »	30
3.3.3	Cas d'utilisation « Gérer les "Profils des animaux domestiques" »	32
3.3.4	Cas d'utilisation « Ajouter une note pour un animal »	35
3.4	Sprint 3 : Gestion des demandes d'accès du vétérinaire	37
3.4.1	Diagramme de cas d'utilisation du troisième sprint	37
3.4.2	Cas d'utilisation « Demande d'ouverture d'un compte vétérinaire »	38
3.4.3	Cas d'utilisation « Gestion des demandes d'ouverture d'un compte vétérinaire »	40
3.4.4	Cas d'utilisation « Modification d'un compte vétérinaire »	42
3.5	Sprint 4 : Messagerie	45
3.5.1	Diagramme de cas d'utilisation du sprint 4	45
3.5.2	Cas d'utilisation « Envoyer des messages »	46
3.5.3	Diagramme de classe du quatrième sprint	48
3.6	Conclusion	48
4	deuxième release	49
4.1	Introduction	49
4.2	Sprint 5 : signalement des alertes	49
4.2.1	Diagramme de cas d'utilisation du sprint 5 :	49
4.2.2	Cas d'utilisation « Signaler des alertes »	50
4.2.3	Cas d'utilisation « Gestion des alertes »	51
4.2.4	Diagramme de classe du sprint 5	52
4.3	Sprint 6 : la prise des rendez-vous	53
4.3.1	Introduction	53
4.3.2	Diagramme de cas d'utilisation du sprint 6	53
4.3.3	Cas d'utilisation « consulter le profil et voir les informations du vétérinaire »	54
4.3.4	Cas d'utilisation « Prendre un rendez-vous »	56
4.3.5	Cas d'utilisation « Gestion des rendez-vous »	58
4.3.6	Diagramme de classe du sprint6	60
4.4	Conclusion	61
5	Troisième Release	62
5.1	Introduction	62
5.2	sprint 7 : « L'adoption »	62
5.2.1	Diagramme de cas d'utilisation du sprint 7	63
5.2.2	Cas d'utilisation « Adopter un animal »	63
5.2.3	Cas d'utilisation « Gestion des adoptions »	65
5.2.4	Diagramme de classe de sprint 7	67

5.3	Sprint 8 : « Publication »	67
5.3.1	Diagramme de cas d'utilisation du sprint 8« Publication »	68
5.3.2	Cas d'utilisation « Effectuer des publications »	68
5.3.3	Cas d'utilisation « signaler des publications »	70
5.3.4	Diagramme de classe de sprint 8	72
5.4	Conclusion	72
6	Modélisation de la Base de données	73
6.1	Introduction	73
6.2	Règles de gestion	73
6.3	Diagramme de classe	74
6.4	Dictionnaire de données et description des classes	75
6.4.1	Schéma relationnel	78
6.4.2	Modèle relationnel	78
6.5	Conclusion	79
7	Réalisation	80
7.1	Introduction	80
7.2	Langages, outils et environnement de développement	80
7.2.1	Côté Backend	80
7.2.2	Côté Frontend	81
7.2.3	environnement de développement	82
7.3	Etude comparative finale	84
7.3.1	Analyse comparative	84
7.4	Présentation des interfaces	85
7.4.1	Interface d'accueil	85
7.4.2	Interface d'accueil	85
7.4.3	Interface d'inscription	85
7.4.4	Interface de connexion	86
7.4.5	Interface Espace propriétaire	87
7.4.6	Interface creation compte animal	88
7.4.7	Interface Espace création compte animal	88
7.4.8	Interface espace liste de mes animaux	89
7.4.9	Interface profil d'un animal	89
7.4.10	Interface de la prise de note	90
7.4.11	Interface de la liste des notes prises	90
7.4.12	Interface Espace publication	91
7.4.13	Interface publication	92
7.4.14	Interface signaler une publication	93
7.4.15	Interface de gestion des signalement des publications	94
7.4.16	Interface messagerie	95
7.4.17	Interface liste des vétérinaire	95
7.4.18	Interface profil du vétérinaire	96
7.4.19	Interface de la liste des demandes de rendez-vous	97
7.4.20	Interface prise de rendez-vous	97
7.4.21	Interface Espace adoption	98
7.4.22	Interface adopter un animal	99
7.4.23	Interface de gestion des adoptions	99
7.4.24	Interface alerte	100
7.4.25	Interface de gestion des alertes	101

7.4.26	Interface demande d'ouverture d'un compte vétérinaire	102
7.4.27	Interface de gestion des demandes d'ouverture de comptes vétérinaire . .	103
7.5	Conclusion	103
8	Conclusion générale	104
9	Bibliographie	105

Table des figures

1.1	Architecture à trois tiers	3
1.2	L'accueil de la plateforme Yummypets	6
1.3	l'application de Wamiz.com	7
2.1	les principes composants de la méthode de Scrum	11
2.2	Types du diagramme UML	13
2.3	Le modèle MVC	14
2.4	Diagramme de contexte	17
2.5	Diagramme de cas d'utilisation	18
2.6	Schéma de la planification des releases	19
2.7	Logo de l'application	21
2.8	Palette de couleur	22
3.1	Diagramme de cas d'utilisation de sprint 1	24
3.2	Diagramme d'interaction de cas d'utilisation « Inscription »	26
3.3	Diagramme d'interaction de cas d'utilisation « se connecter»	28
3.4	Diagramme de classe« Sprint 1»	29
3.5	Diagramme de cas d'utilisation « sprint2 »	30
3.6	Diagramme d'interaction de cas d'utilisation « créer Profil animal»	32
3.7	Diagramme d'interaction de cas d'utilisation « modifier compte animal»	34
3.8	Diagramme d'interaction de cas d'utilisation «Supprimer animal »	35
3.9	Diagramme d'interaction de cas d'utilisation « Ajouter une note pour un animal»	36
3.10	Diagramme de classe du sprint02»	37
3.11	Diagramme de cas d'utilisation « Demande d'ouverture d'un compte vétérinaire»	38
3.12	Diagramme d'interaction de « demande d'ouverture un compte vétérinaire»	40
3.13	Diagramme d'interaction de « gestion des demandes d'ouverture comptes vétérinaires»	42
3.14	diagramme d'interaction modifier compte vétérinaire	44
3.15	diagramme de classe sprint 3	45
3.16	Diagramme d'interaction de cas d'utilisation « Envoyer des messages»	46
3.17	Diagramme d'interaction de cas d'utilisation « Envoyer des messages»	47
3.18	Diagramme de classe du sprint4.	48
4.1	Diagramme de cas d'utilisation du sprint 5.	50
4.2	Diagramme d'intéraction de cas d'utilisation 'Signaler des alertes'	51
4.3	Diagramme d'intéraction de cas d'utilisation 'Gestion des alertes'	52
4.4	diagramme de classe sprint 05	53
4.5	Diagramme de cas d'utilisation du sprint6.	54
4.6	Diagramme d'interaction de cas d'utilisation « rechercher un vétérinaire ».	56
4.7	Diagramme d'interaction de cas d'utilisation prendre rendez vous	58
4.8	Diagramme d'interaction de cas d'utilisation gérer les demandes de rendez-vous	60

4.9	Diagramme de classe du sprint6	61
5.1	Diagramme de cas d'utilisation du sprint 7	63
5.2	Diagramme d'interaction du cas d'utilisation « Adopter un animal »	65
5.3	Diagramme d'interaction du cas d'utilisation « Gestion des adoptions »	66
5.4	Diagramme de classe « sprint7 »	67
5.5	Diagramme de cas d'utilisation du sprint 8	68
5.6	Diagramme d'interaction du cas d'utilisation « Effectuer des publications »	69
5.7	Diagramme d'interaction du cas d'utilisation «signaler la publication»	71
5.8	Diagramme de classe «sprint8»	72
6.1	Diagramme de classe	75
7.1	Interface d'accueil de petracker (partie 1)	85
7.2	Interface d'accueil de petracker (partie 2)	85
7.3	Interface d'inscription de petracker	86
7.4	interface de connexion	87
7.5	Interface d'accueil de l'espace propriétaire	88
7.6	Interface de création d'un compte animal partie 1	88
7.7	Interface de creation d'un compte animal partie2	89
7.8	Interface liste de mes animaux	89
7.9	Interface profil animal	90
7.10	Interface prise note pour animal	90
7.11	Interface note pour animal	91
7.12	Interface publier	91
7.13	Interface publication	92
7.14	signaler la publication	93
7.15	signaler la publication	94
7.16	Interface messagerie	95
7.17	Interface liste vétérinaire	95
7.18	Interface profil du vétérinaire	96
7.19	Interface liste des rendez-vous	97
7.20	Interface prise de rendez-vous	98
7.21	Interface adoption	98
7.22	Interface adopter un animal	99
7.23	Interface gestion des demandes d'adoptions	100
7.24	Interface alerte	100
7.25	Interface gestion des alertes	101
7.26	formulaire de demande d'ouverture d'un compte vétérinaire	102
7.27	Interface gestion des demandes d'ouverture de compte vétérinaire	103

Liste des tableaux

1.1	Comparaison des fonctionnalités des applications	8
2.2	Répartition des rôles Scrum	14
2.3	User Stories	15
2.4	Le Backlog de notre futur système	21
3.1	Description du cas d'utilisation pour l'inscription	25
3.2	Description textuelle du cas d'utilisation « Se connecter »	27
3.3	Description textuelle du cas d'utilisation « Créer des comptes pour les animaux domestiques »	31
3.4	la description de cas d'utilisation Gérer les Profils des animaux	33
3.5	Description textuelle du cas d'utilisation « Ajouter une note pour un animal »	36
3.6	Description textuelle du cas d'utilisation « Demande d'ouverture d'un compte vétérinaire »	39
3.7	la description de cas d'utilisation Gestion des demandes d'ouverture d'un compte vétérinaire	41
3.8	la description de cas d'utilisation modification compte veterinaire	43
3.9	la description de cas d'utilisation Envoyer des messages	47
4.1	la description de cas d'utilisation signaler alerte	50
4.2	la description de Cas d'utilisation « Gestion des alertes »	52
4.3	la description de cas d'utilisation « consulter le profil et voir les informations du vétérinaire »	55
4.4	description de cas d'utilisation « prendre un rendez vous »	57
4.5	description de cas d'utilisation gestion rendez-vous	59
5.1	description de cas d'utilisation adopter animal"	64
5.2	description de cas d'utilisation « Gestion des adoptions »	66
5.3	description de Cas d'utilisation « Effectuer des publications »	69
5.4	description de Cas d'utilisation « signaler des publications »	70
6.1	Dictionnaire de données	77
7.1	Comparaison des fonctionnalités des applications	84

Liste des Abréviations

- **WWW** : World Wide Web
- **HTML** : HyperText Markup Language
- **HTTP** : HyperText Transfer Protocol
- **URL** : Uniform Resource Locator
- **UML** : Unified Modeling Language
- **SQL** : Structured Query Language
- **CRUD** : Create, Read, Update, Delete
- **API** : Application Programming Interface
- **MVC** : Model-View-Controller (Modèle-Vue-Contrôleur)
- **CU** : Cas d'Utilisation
- **Git** : Git hub
- **RDV** : Rendez-vous
- **NB** : Remarque
- **Enum** : Énumération
- **ID** : Identifiant
- **PHP** : HyperText Preprocessor
- **ORM** : Object-Relational Mapping
- **DOM** : Document Object Model
- **AJAX** : Asynchronous JavaScript and XML
- **VS Code** : Visual Studio Code

Introduction générale

Le domaine de l'adoption et du sauvetage des animaux domestiques est confronté à plusieurs défis, notamment le nombre croissant d'animaux abandonnés, et la difficulté de leurs trouver des foyers permanents. Chaque année, des milliers d'animaux domestiques se retrouvent sans abri, souvent victimes de circonstances imprévisibles. Ces animaux nécessitent non seulement des soins et de l'attention, mais aussi une nouvelle famille capable de leur offrir une vie meilleure. C'est dans ce contexte que notre application 'Pettracker' se positionne comme solution innovante pour répondre à ces problématiques en facilitant l'adoption et en centralisant les efforts de sauvetage.

Au cours de ce projet, nous avons analysé les obstacles rencontrés par les refuges et les associations de protection animale et les potentiels adoptants. Les défis principaux incluent :

- La complexité et la longueur du processus de l'adoption
- L'absence d'un système efficace de suivi et de gestion des rendez-vous pour les visites et les rencontres.
- La nécessité d'un mécanisme d'alerte pour les situations d'urgence comme la perte d'un animal, et les compagnes de sauvetage.

En réponse à ces défis, nous avons conçu et développé 'Pettracker', une application web qui vise à faciliter le processus d'adoption et de sauvetage des animaux de compagnie. Pettracker offre une plateforme centralisée en mettant en avant les profils détaillés des animaux. L'application permet également la prise de rendez-vous en ligne, simplifiant ainsi la planification des visites et des consultations avec des vétérinaires. De plus, Pettracker intègre un système d'alerte qui informe les utilisateurs des situations d'urgence et des événements de sauvetage, renforçant ainsi l'engagement et la mobilisation communautaire.

En résumé, Pettracker ambitionne de transformer le processus d'adoption et de sauvetage des animaux domestiques en le rendant plus efficace. La suite de ce rapport est structurée comme suit : Le premier chapitre introduit le projet en présentant les principaux concepts du web, les défis locaux de gestion des animaux domestiques, les objectifs innovants de notre application, et les lacunes des plateformes existantes qui ont inspiré son développement. Le deuxième chapitre décrit la conception méthodologique et les spécifications du projet via la méthode agile SCRUM, l'utilisation de UML pour modéliser les composants, et l'architecture MVC pour structurer le système. Le troisième, le quatrième et le cinquième chapitre détaillent les phases de développement et d'implémentation de l'application, y compris les étapes de conception et les livrables associés. Le sixième chapitre représente la base de données et le schéma relationnel. Le septième chapitre est consacré à l'implémentation de l'application en présentant les langages, les outils et les environnements de développement utilisés. Par la suite, nous présenterons quelques interfaces utilisateur. Enfin, nous clôturons ce mémoire par une conclusion générale qui résume les résultats obtenus et propose des perspectives d'évolution future pour Pettracker.

Chapitre 1

Contexte d'étude et recueil des besoins

1.1. Introduction

Dans ce chapitre, nous introduisons le contexte et les objectifs de notre projet Petracker. Nous débuterons par une introduction aux concepts fondamentaux du web, suivi d'une analyse des défis associés à la gestion des animaux domestiques au niveau local. Ensuite, nous présentons nos objectifs et des solutions novatrices que nous proposons, tout en examinant les facteurs clés de succès de notre application. Enfin, nous identifions les lacunes actuelles des plateformes existantes qui ont motivé le développement de cette solution innovante.

1.2. Concepts fondamentaux du web

1.2.1. Fondamentaux des Technologies Web

Les technologies web constituent l'ensemble des outils et des techniques utilisés pour la création et l'interaction avec les sites web. Elles comprennent les langages de programmation, les protocoles et les standards qui permettent aux développeurs de concevoir des pages web interactives et dynamiques.

1.2.2. Concepts fondamentaux du web

- **Page web** : document accessible via un navigateur web, utilisant principalement le langage HTML pour le contenu [1].
- **Site web** : est un ensemble de pages web interconnectées, hébergées sous un nom de domaine commun et disponibles au public et souvent développé pour divers objectifs [1].
- **Navigateur web** : est un logiciel permettant aux utilisateurs d'accéder et d'interagir avec des pages web. Parmi les navigateurs les plus populaires, on trouve Google Chrome, Mozilla Firefox, Microsoft Edge [1].
- **HTTP (Hyper Text Transfer Protocol)** : un protocole de communication entre les clients (tels que les navigateurs web) et les serveurs web pour récupérer et afficher des contenus en ligne [1].
- **URL (Uniform Resource Locator)** : adresse web unique indiquant l'emplacement précis d'une ressource sur Internet [1].

1.2.3. Classification des sites web

Il existe deux types principaux de sites web :

- **Les sites web statiques** : un site web statique est composé de pages web dont le contenu reste fixe, indépendamment de l'utilisateur qui y accède. Ce type de site est rapide à charger et facile à héberger, car il ne nécessite pas de serveur de base de données. La création d'un site web statique est souvent moins coûteuse qu'un site web dynamique, ce qui en fait une option économique pour de nombreux projets[2].
- **Les sites web dynamiques** : les sites web dynamique, en revanche, génèrent du contenu en temps réel en fonction des demandes des utilisateurs. Ils sont plus interactifs et permettent une personnalisation plus avancée, mais nécessitent un serveur de base de données et des technologies plus complexes pour fonctionner, ce qui rend leur développement et leur maintenance plus coûteux et exigeant en termes de ressources[2].

1.2.4. Architecture à trois tiers

L'architecture à trois tiers est un modèle avancé couramment utilisé dans le développement des applications web. Ce modèle sépare l'application en trois couches distinctes :

- **Tier de présentation (Frontend)** : cette couche gère l'interface utilisateur et les interactions avec celui-ci.
- **Tier de logique métier (Backend)** : cette couche contient la logique métier de l'application et traite les données en fonction des règles définies.
- **Tier de données** : gère l'accès et la gestion des données, souvent en interaction avec une base de données [3].

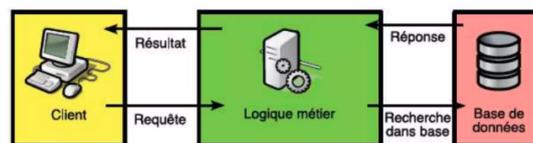


fig 1.1 : Architecture à trois tiers

1.3. Contexte de l'adoption et du sauvetage des animaux domestiques

La relation entre les humains et les animaux domestiques est profonde et diversifiée, englobant la simple compagnie ainsi que des rôles essentiels dans la sécurité, la thérapie, et le sauvetage. En Algérie, cette proximité soulève des défis particuliers dans la gestion du bien-être des animaux, notamment en ce qui concerne l'adoption, les rendez-vous vétérinaires, et la réaction aux urgences.

1.3.1. Problématique

1. Enjeux actuels en Algérie

- **Absence d'application centralisée pour l'adoption et le sauvetage** : actuellement, il n'existe pas d'application dédiée à l'adoption et au sauvetage des animaux domestiques en Algérie, mis à part quelques initiatives locales et animaleries. Cette absence de plateforme centralisée complique le processus d'adoption et de sauvetage, rendant difficile pour les potentiels adoptants de trouver des animaux à adopter et pour les refuges et les particuliers de diffuser leurs annonces d'adoption.

- **Gestion désorganisée des soins vétérinaires et des urgences** : La gestion des soins vétérinaires et la réponse aux urgences se fait souvent de manière désorganisée, sans outil numérique pour faciliter ces tâches essentielles. Une application dédiée pourrait améliorer l'interaction entre les vétérinaires, les refuges et les propriétaires d'animaux.
- **Croissance de la sensibilisation à la cause animale** : Heureusement, la sensibilisation à la cause animale est en croissance en Algérie. Des initiatives locales et des campagnes de sensibilisation, souvent menées par des associations de protection animale, visent à promouvoir l'adoption et à réduire l'abandon des animaux. Ces efforts montrent un besoin croissant pour des solutions innovantes et accessibles pour améliorer le bien-être des animaux domestiques et faciliter leur adoption.

2. La pratique de « Galoufa »

En Algérie, la pratique de "Galoufa" est malheureusement répandue dans certaines régions. Ce terme désigne l'empoisonnement délibéré des animaux errants comme une méthode de contrôle de leur population. Cette pratique est souvent perçue comme une solution rapide et peu coûteuse pour gérer le problème des animaux errants, mais elle soulève de graves préoccupations éthiques et environnementales. L'usage de poisons pour éliminer les animaux errants est non seulement cruel et inhumain, mais aussi inefficace à long terme. Pour mieux comprendre l'origine et l'impact de cette pratique, Sarah Laadjal précise dans son article du 26 septembre 2018 [4] :

« La chasse aux animaux appelée « Galoufa », est une pratique qui date de la période coloniale. Le premier « capteur » de chiens de la ville d'Alger entra dans la légende et donna même son nom, déformé par le langage populaire, à tous ses futurs confrères d'Algérie. D'origine espagnole, Garufa, tout de suite appelé Galoufa, habitait le quartier de Bab el-oued. »

Malheureusement, cette pratique continue à se perpétuer à Alger, ciblant principalement les chiens et les chats. Les méthodes utilisées sont particulièrement cruelles, comme le souligne Haifa Rezagui, vétérinaire et défenseuse des animaux [5] :

« Les animaux sont électrocutés, alors que cette méthode est interdite par la loi, car elle entraîne une mort lente et douloureuse pour l'animal. Les animaux sont aspergés d'eau puis électrocutés et la mort survient une demi-heure plus tard sous un arrêt cardiaque. »

En plus de causer des souffrances inutiles aux animaux, elle présente des risques pour la santé publique et l'environnement. Les substances utilisées peuvent contaminer l'écosystème local et poser des dangers pour d'autres animaux, y compris les animaux de compagnie et la faune sauvage.

1.3.2. Rôle de « Petracker » dans ce contexte

C'est dans ce contexte critique que « Petracker » se positionne comme une solution novatrice et humanitaire. En offrant une plateforme numérique dédiée à l'adoption et au sauvetage des animaux domestiques en Algérie, notre application crée une communauté unie et proactive. Cette communauté est capable de répondre aux défis de la gestion des animaux domestiques et errants de manière plus efficace et humaine, jouant un rôle crucial dans la réduction de la pratique de "Galoufa". De plus, en sensibilisant la population à travers des campagnes éducatives et des partages communautaires, nous encourageons une approche plus empathique envers les animaux domestiques. « Petracker » promeut ainsi une cohabitation harmonieuse entre les humains et les animaux en Algérie.

1.3.3. Solutions proposées et mission de l'application

Pour répondre à ces défis, nous proposons une plateforme qui vise à atteindre les objectifs suivants :

- **Faciliter l'adoption et le sauvetage des animaux domestiques** : Grâce à une plateforme centralisée, "Pettracker" permet aux adoptants potentiels de trouver facilement des animaux disponibles et aux refuges et particuliers de diffuser efficacement leurs annonces.
- **Améliorer la gestion des soins vétérinaires** : L'application offre une fonctionnalité de prise de rendez-vous avec les vétérinaires, simplifiant la planification et la gestion des soins médicaux pour les animaux domestiques.
- **Réagir efficacement aux urgences** : Un espace dédié aux alertes d'urgence permet de signaler rapidement les situations critiques, assurant une réponse rapide et coordonnée pour le sauvetage des animaux.
- **Sensibiliser la population à la cause animale** : Par le biais de l'espace de publication communautaire, les utilisateurs peuvent partager des contenus liés à l'adoption, au sauvetage et à d'autres sujets pertinents, renforçant la sensibilisation et l'engagement de la communauté.

1.4. Facteurs clés du succès de l'application

Pour assurer le succès de notre application, plusieurs facteurs clés doivent être pris en compte :

- **Convivialité de l'interface utilisateur** : L'application doit être intuitive et facile à utiliser, avec une conception axée sur l'expérience utilisateur.
- **Sécurité des données personnelles des utilisateurs** : La protection des données utilisateurs est primordiale. Notre application doit garantir la confidentialité et la sécurité des informations personnelles des utilisateurs, y compris les détails de leurs animaux et les communications avec les vétérinaires.
- **Collaboration étroite avec les organisations de protection animale** : Le succès de l'application repose sur des partenariats solides avec les refuges, les cliniques vétérinaires et les associations de protection animale. Ces collaborations assurent une mise à jour régulière des profils d'animaux disponibles à l'adoption, d'améliorer les services de soins, et de promouvoir des campagnes de sensibilisation efficaces.
- **Sensibilisation et éducation** : La réussite de « Pettracker » dépend de sa capacité à éduquer et à sensibiliser le public sur l'importance de l'adoption et bien-être animal. Des campagnes éducatives régulières et des contenus informatifs peuvent aider à changer les attitudes et à promouvoir des pratiques plus humaines.

1.5. Plateformes existantes et besoins non comblés

Dans le cadre de notre projet visant à développer une application dédiée aux animaux de compagnie, nous avons entrepris une analyse approfondie des solutions actuellement disponibles à l'échelle internationale ainsi que des applications situées principalement en France. Cette démarche vise à identifier les points forts et les faiblesses des plateformes similaires existantes, afin de déterminer les besoins spécifiques du marché local et d'orienter notre propre développement.

1.5.1. Étude des applications existantes

Après avoir effectué des recherches approfondies, nous avons recensé plusieurs applications pertinentes dans notre domaine d'intérêt :

I. Etude du site Yummypets [6]

- **Nom du site web** : Yummypets.
- **URL** : <https://fr.yummypets.com/user/board>
- **Accès au site** : Inscription requise, avec ajout de l'animal de compagnie.
- **Catégorie de l'annonce** : Multiples.
- **Recherche** : Rapide par mot clés ou avancée.
- **Rôle** : Plateforme en ligne pour les passionnés des animaux.

Points positifs (+) :

- Vaste catalogue d'annonces grâce à l'ancienneté de la plateforme (créé en 2012). Encourage l'adoption et sensibilise à la cause animale.
- Variété de contenu : Articles, photos, vidéos et témoignages couvrent une multitude d'animaux et de sujets.
- Mise en place d'un espace personnel pour les animaux permettant la création de leurs profils et le partage de leurs aventures.
- Propose des informations sur la santé, l'alimentation et le bien-être animal.

Points négatifs (-) :

- Absence de mécanisme de signalement pour les contenus inappropriés.
- Absence de messagerie privée entre les utilisateurs.
- Manque d'interaction avec les vétérinaires ce qui complique aux utilisateurs la prise de rendez-vous directe ou la réception des conseils médicaux.
- Manque de filtres de recherches avancés des animaux en fonction de critères spécifiques.



fig 1.2 : L'accueil de la plateforme Yummypets

II. Etude du site Wamiz [7]

- **Nom du site web** : Wamiz.

- **URL** : <https://Wamiz.com/>
- **Accès au site** : Inscription requise pour avoir un accès complet au contenu du site.
- **Catégorie de l'annonce** : Le site propose différentes catégories d'annonces liées aux animaux de compagnie.
- **Rôle** : Site de conseils et d'actualité créée en 2009, guide les propriétaires et futurs propriétaires d'animaux de compagnie grâce à un contenu complet et expert. Il propose des informations sur l'adoption, l'éducation, la santé ainsi que des actualités, quizz et jeux, attirant ainsi tous les amoureux des animaux.

Points positifs (+) :

- Annonces d'adoption et de recherche d'animaux.
- Informations complètes sur les races des animaux domestiques.
- Actualités et divertissements pour les amoureux des animaux.
- Conseils d'experts en comportement et santé animale

Points négatifs (-) :

- Manque de fonctionnalités pour l'interaction sociale entre les propriétaires d'animaux.
- Manque de fonctionnalité pour le suivi des vaccins et visites chez les vétérinaires.
- Manque de localisation des services vétérinaires

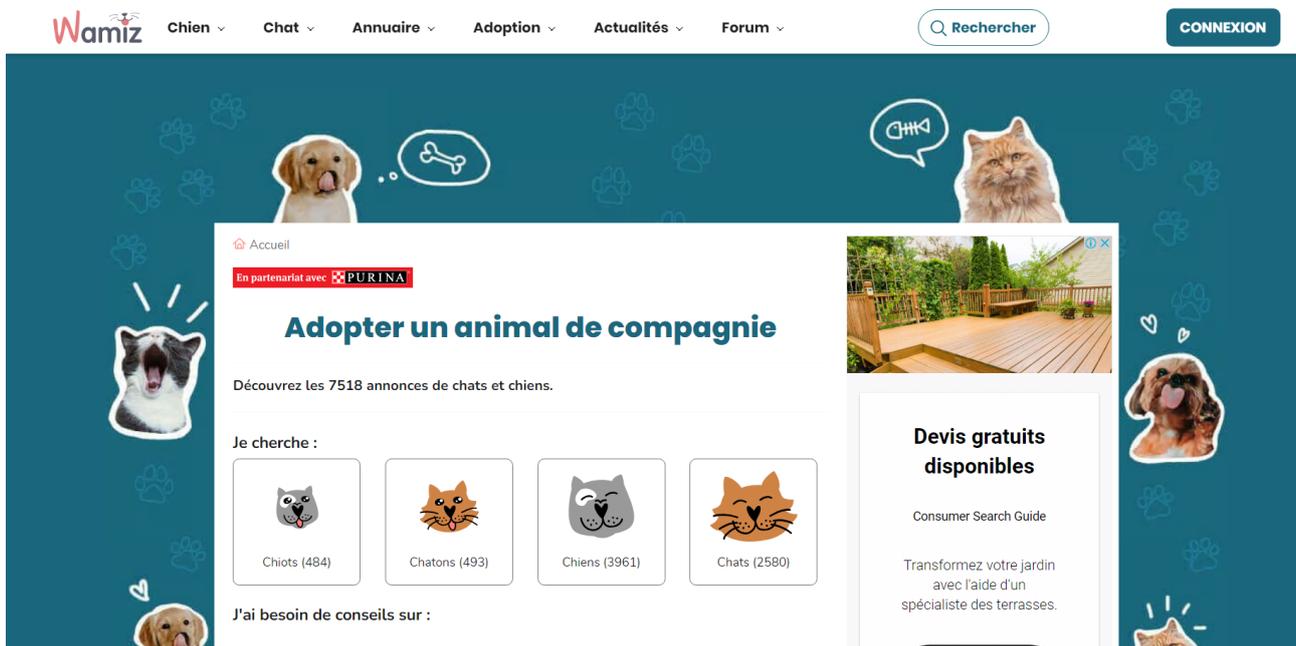


fig 1.3 : l'application de Wamiz.com

1.5.2. Étude comparative

Pour réaliser une étude comparative entre les applications Wamiz, Yummypets et notre application, nous allons créer un tableau qui met en évidence les différentes fonctionnalités offertes par chaque application.

Fonctionnalité	Wamiz	Yummypets
Espace de publication	✓	✓
Espace d'alerte d'urgence	✗	✗
Espace Messagerie	✓	✗
Espace rendez-vous	✗	✗
Espace Adoption	✓	✓
Espace de création des comptes animaux	✗	✓
Mini agenda	✗	✗
Facilité d'utilisation	Bonne	Bonne
Disponibilité (web/mobile)	Web et mobile	Web et mobile

TABLE 1.1 – Comparaison des fonctionnalités des applications

1.5.3. Description des Fonctionnalités

- **Espace Publication** : Permet aux utilisateurs de publier du contenu multimédia (photo, vidéo) et mises à jour sur leurs animaux de compagnie.
- **Espace Alerte d'Urgence** : Fonctionnalité permettant de signaler et coordonner les secours pour les animaux en détresse.
- **Espace Messagerie** : Système de messagerie interne permettant aux utilisateurs de communiquer entre eux.
- **Espace Rendez-vous** : Fonctionnalité permettant de planifier des rendez-vous avec des vétérinaires.
- **Espace Adoption** : Section dédiée à l'adoption des animaux avec des profils détaillés des animaux disponibles.
- **Espace de Création des Comptes Animaux** : Permet aux propriétaires des animaux de leur créer des profils détaillés.
- **Mini Agenda** : Un petit agenda intégré pour gérer les événements et rappels liés aux soins et activités des animaux.
- **Facilité d'Utilisation** : Mesure de la convivialité et de la simplicité d'utilisation de l'application.
- **Disponibilité (Web/Mobile)** : Indique sur quelles plateformes l'application est disponible (web et/ou mobile).

1.5.4. Analyse comparative

- **Wamiz** : Propose des fonctionnalités de publication, messagerie et adoption, mais manque d'alertes d'urgence, de gestion des rendez-vous, de création de comptes animaux et de mini agenda. Disponible sur web et mobile.
- **Yummypets** : Offre des fonctionnalités de publication, d'adoption et de création de comptes animaux, mais ne propose pas d'alertes d'urgence, de messagerie entre utilisateurs, de gestion des rendez-vous ni de mini agenda. Disponible sur web et mobile.

1.6. Conclusion

Ce premier chapitre a établi les bases nécessaires pour comprendre le contexte et les besoins de notre projet Petracker. Nous avons exploré les défis locaux significatifs, tels que l'absence

d'une plateforme centralisée et les pratiques préjudiciables comme la « Galoufa ». Petracker s'engage à répondre à ces défis en offrant une solution innovante qui facilite l'adoption, améliore la gestion des soins vétérinaires, réagit efficacement aux urgences et sensibilise la population à la cause animale. De plus, notre analyse comparative des applications existantes telles que Yummypets et Wamiz a permis d'identifier clairement les forces et les faiblesses des solutions actuellement disponibles. Ces informations sont cruciales pour guider le développement de Petracker, en nous aidant à adopter les meilleures méthodes et à répondre efficacement aux besoins particuliers du marché algérien. En combinant ces éléments, nous sommes mieux préparés pour aborder les phases suivantes de notre projet avec une compréhension approfondie des défis à surmonter. Dans le prochain chapitre nous aborderons la méthodologie de la conception et spécification des besoins.

Chapitre 2

Méthodologie de la conception et spécification des besoins

2.1. Introduction

Dans ce chapitre, nous allons présenter la méthodologie de la conception et la spécification des besoins, nous allons présenter d'abord la méthode agile (Scrum), nous expliquerons comment appliquer la méthode Scrum sur notre projet, Nous utiliserons le langage de modélisation UML (Unified Modeling Language) pour représenter les différentes composantes de notre système. Ensuite, nous allons aborder l'architecture MVC (Model View Controller), en expliquant comment cette architecture structure notre système.

2.2. Méthodologie de la conception

Pour l'élaboration de notre projet, nous avons fait recours aux méthodes de conceptions agiles et langage de modélisation unifié UML.

2.2.1. Qu'est-ce qu'une Méthode agile ?

La méthode Agile est une approche de gestion de projet qui valorise la flexibilité, la collaboration et la réponse rapide aux changements. Elle encourage le cycle de développement courts et fréquents pour une livraison continue de valeur, centrée sur l'humain, elle implique des clients dans le processus de création pour mieux répondre à leur besoin contrairement aux méthodes traditionnelles qui suivent un processus linéaire et séquentiel, avec les phases de projet bien définies [8].

La méthode agile repose sur plusieurs principes

- La satisfaction de client : ce principe place le client au cœur du processus de développement, l'objectif est de livrer rapidement des fonctionnalités qui apportent la valeur, en mettant l'accent sur l'écoute et la réponse au retour de client pour garantir leur satisfaction. Ce qui implique la mise sur le marché accélérée des produits et une capacité à intégrer les feedbacks pour améliorer continuellement le produit[8].
- Accueil de changement : La méthode agile reconnaît que les besoins des clients peuvent évoluer avec le temps, ce principe souligne l'importance d'être flexible et réceptif au changement même en fin de cycle de développement, pour offrir un produit qui répond précisément aux exigences actuelles du client, lui conférant ainsi un avantage compétitif[8].

- Livraison fréquente : l'accent est mis sur la livraison continue de segment de travail fonctionnels, ce qui permet aux clients de voir régulièrement le progrès et d'adopter des ajustements si nécessaire, les cycle de développement, les cycles de développement courts encouragent une amélioration constante du produit et une réduction du temps entre la mise à jour, favorisant ainsi une meilleure réactivité aux demandes du marché [9].
- Collaboration étroite : ce principe valorise la collaboration quotidienne entre de développeur et des clients il reconnaît que les solutions les plus efficaces et les plus innovantes proviennent d'équipe autoorganisées qui communiquent ouvertement et travaillent en étroite synergie avec client, cette approche renforce l'engagement et l'alignement sur les objectifs communs du projet [9].

2.2.2. Méthode Scrum

Scrum est une structure agile qui soutient les équipes de leur collaboration et les guide vers l'accomplissement efficaces de taches importantes, il fournit un ensemble de valeurs, de rôles et de règles qui aident des équipes à se focaliser sur l'amélioration continue à travers des itérations régulières. Dans Scrum le développement est structuré en sprints, des périodes de travail intensif de deux à quatre semaines, au terme desquelles un produit fonctionnel est attendu ,deux autre rituels clés structurent le processus Scrum, les mêlées quotidienne et les rétrospective de sprint, les mêlées quotidiennes réunis brèves de synchronisation, permettent à l'équipe de planifier la journées en 15 minutes ,les rétrospectives de sprint, dirigées par le Scrum Master ,offrent un moment de réflexion collective à la fin de chaque sprint ,dirigées par le Scrum Master, offrent un moment de réflexion collective à la fin de chaque sprint pour évaluer la performances et planifier des améliorations futur[12].

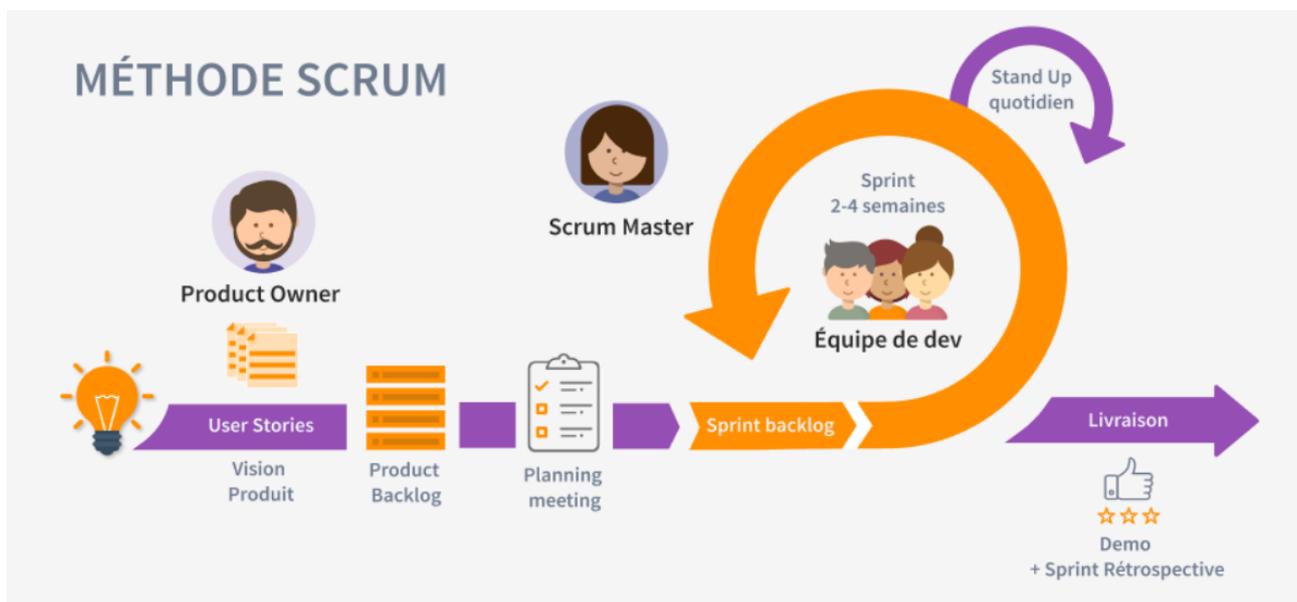


fig 2.1 : les principes composants de la méthode de Scrum

Les parties-prenantes

la méthode Scrum regroupe trois acteurs [13] :

- **Le Product Owner** : représente l'utilisateur ou le propriétaire.
- **L'équipe de développement** : est composée de professionnels réalisant le travail, ils gèrent directement le sprint et le Backlog, et s'assurent que le produit est développé selon les critères de qualité définis.

- **Le Scrum master** : agit comme un coach pour l'équipe Scrum, aidant à comprendre et à mettre en œuvre Scrum et ses éléments, il travaille à éliminer les obstacles qui pourraient empêcher l'équipe de développement ou le Product Owner de réaliser leur travail efficacement.

Product Backlog (Carnet de Produit)

Le *Product Backlog* est une liste dynamique et priorisée de tout ce qui pourrait être nécessaire pour le développement complet du produit. Il se présente sous la forme de courtes descriptions appelées *user stories*. Il représente l'ensemble des besoins des clients ou des utilisateurs finaux pour le produit, et est priorisé par le *Product Owner* [14].

Les Sprints

Le *Sprint* est une période de temps délimitée, généralement de 1 à 4 semaines, pendant laquelle l'équipe Scrum travaille pour accomplir un ensemble spécifique de tâches tirées du *Product Backlog*. C'est une période intense visant à produire un incrément de produit potentiellement livrable, qui est ensuite évalué lors de la revue de sprint. Les sprints permettent une adaptation rapide aux changements et une amélioration continue grâce à des cycles de développement courts et réguliers [15].

Release

Release est une version du produit regroupant plusieurs fonctionnalités développées et prêtes à être utilisées par les clients. Elle est planifiée pour apporter une valeur ajoutée significative, souvent alignée avec des objectifs stratégiques ou des jalons de projet. Ce processus permet une amélioration continue du produit en intégrant régulièrement les retours des utilisateurs [16].

2.2.3. Langage de modélisation UML

Dans notre projet nous nous appuyerons sur langage de UML (Unified Modeling Language) (en français langage de modélisation unifié) qui est un langage de modélisation graphique standardisé pour la visualisation, la spécification, la construction et la documentation des artefacts d'un système informatique. Il représente l'aspect structurel d'un programme via d'un diagramme, bien qu'il ne soit pas un langage de programmation il facilite la génération de code dans plusieurs langages, il est largement utilisé dans développement de logiciel et en conception orienté objet. Il existe deux types de vues de système : Les Diagrammes structurels et les diagrammes comportementaux comme le montre la figure suivante[20] :

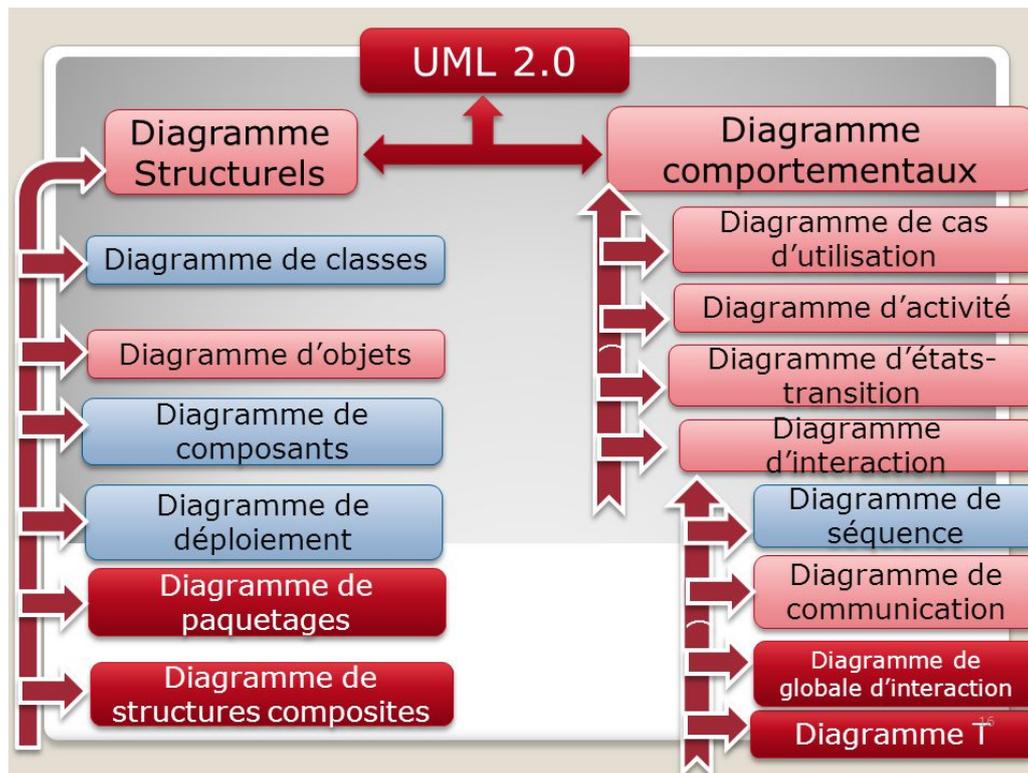


fig 2.2 : Types du diagramme UML

2.2.4. Modèle MVC

Le *Modèle-Vue-Contrôleur* ou *MVC* est une architecture logicielle très connue et largement utilisée pour organiser l'interface graphique d'un programme. Il consiste à distinguer trois entités distinctes :

Modèle Élément qui contient les données ainsi que la logique en rapport avec les données : validation, lecture et enregistrement. Il peut, dans sa forme la plus simple, contenir uniquement une simple valeur ou une structure de données plus complexe. Le modèle représente l'univers dans lequel s'inscrit l'application. Par exemple, pour une application de banque, le modèle représente des comptes, des clients, ainsi que les opérations telles que dépôt et retraits, et vérifie que les retraits ne dépassent pas la limite de crédit [18].

Vue Partie visible d'une interface graphique. La vue se sert du modèle, et peut être un diagramme, un formulaire, des boutons, etc. Une vue contient des éléments visuels ainsi que la logique nécessaire pour afficher les données provenant du modèle. Dans une application de bureau classique, la vue obtient les données nécessaires à la présentation du modèle en posant des questions. Elle peut également mettre à jour le modèle en envoyant des messages appropriés. Dans une application web, une vue contient des balises HTML[18].

Contrôleur Module qui traite les actions de l'utilisateur, modifie les données du modèle et de la vue [18].

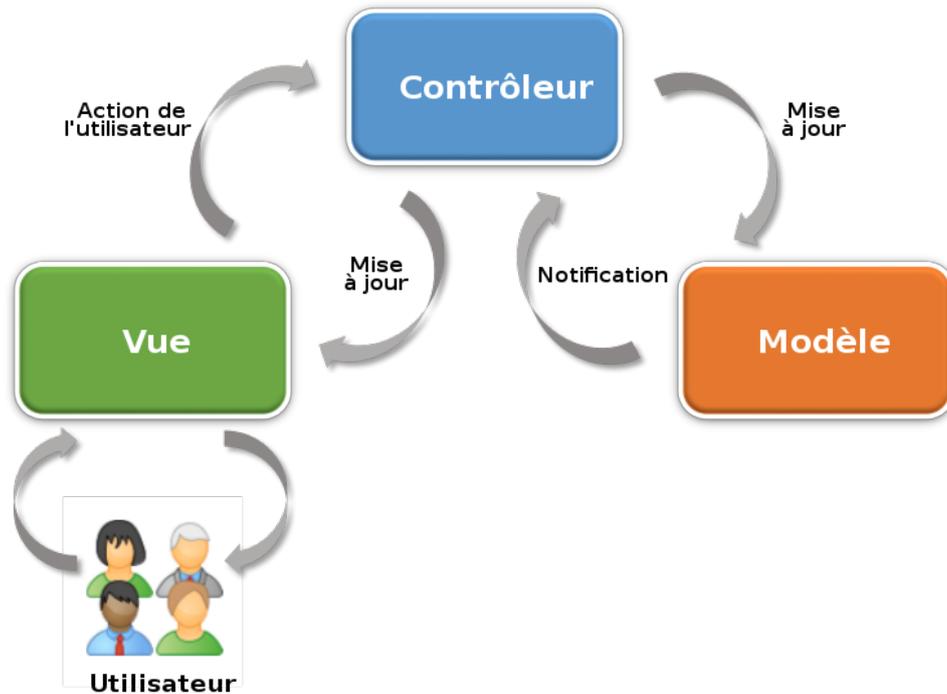


fig 2.3 : Le modèle MVC

2.3. Pilotage de projet avec Scrum

2.3.1. Rôle et user stories

Pour notre projet, les rôles sont répartis comme suit :

Rôle Scrum	Personnes affectées
Product Owner	Les personnes intréssées par les animaux
Scrum Master	Mr. Zerarga Loutfi
Équipe de développement	Asma Sellami et Melissa Laidi

TABLE 2.2 – Répartition des rôles Scrum

2.3.2. User Stories

Nous présentons sur la figure suivante des user stories qui sont utilisées pour spécifier des besoins et des exigences des utilisateurs finaux du produit et les décomposer en tâches concrètes et réalisables pour l'équipe de développement

En tant que	Je veux
Visiteur	<ul style="list-style-type: none"> — voir les alertes et les dernières publications.
Utilisateur	<ul style="list-style-type: none"> — M'inscrire et accéder à mon espace personnel. — Envoyer des messages. — Publier des publications et interagir avec elles via des "j'aimes" et des commentaires.
Propriétaire	<ul style="list-style-type: none"> — Créer les comptes de mes animaux. — Ajouter des notes pour chacun de mes animaux afin de suivre leur état de santé et leur comportement. — Rechercher les vétérinaires selon leur localisation, et consulter leurs profils pour voir leurs informations et pouvoir les contacter. — Signaler des alertes en cas d'urgence. — Prendre un rendez-vous chez un vétérinaire. — Mettre mes animaux en adoption et gérer les demandes correspondantes. — Adopter des animaux. — Signaler les publications inappropriées.
Adoptant	<ul style="list-style-type: none"> — consulter ou voir la liste des animaux . — Adopter des animaux proposés à l'adoption. — Signaler les publications inappropriées.
Vétérinaire	<ul style="list-style-type: none"> — M'inscrire et soumettre mes informations professionnelles pour pouvoir accéder à mon espace. — Gérer mon compte. — Gérer les demandes de rendez-vous (accepter/refuser). — Signaler les publications inappropriées.
Administrateur	<ul style="list-style-type: none"> — Gérer les demandes d'accès des vétérinaires en approuvant ou en supprimant leurs demandes en fonction de leurs informations professionnelles. — Gérer (accepter, refuser) les alertes. — Gérer (accepter, refuser) les signalements de publications.

TABLE 2.3 – User Stories

2.3.3. Identification des acteurs

Un acteur dans un diagramme de cas d'utilisation représente une entité externe, comme un utilisateur ou un système, qui interagit avec le système étudié. Chaque acteur est associé à au moins un cas d'utilisation spécifique, décrivant comment l'acteur utilise le système pour accomplir un objectif particulier. Les acteurs aident à définir les limites du système en identifiant les sources d'entrée et les destinataires de sortie des interactions système [19].

Visiteur

représente toute personne qui utilise notre application sans être authentifiée. Il a accès aux alertes d'urgence ainsi qu'à quelques actualités concernant les publications sur les animaux.

Utilisateur

représente toutes les personnes qui sont inscrites dans l'application, il effectue des tâches communes telles que l'envoi de messages et publier des publications et interagir avec elles via des j'aimes et des commentaires.

Propriétaire

est une personne qui possède un ou plusieurs animaux domestiques et utilise l'application pour leurs créer des comptes et gérer leurs informations. Il peut prendre des rendez-vous avec des vétérinaires, signaler des alertes d'urgence comme dans le cas d'une perte d'un animal, proposer des animaux à adopter comme il peut lui-même en adopter. Il peut également signaler des publications.

Adoptant

est une personne souhaitant adopter des animaux dans la plateforme ,il peut aussi signaler des publications.

Vétérinaire

est un professionnel de la santé animal qui s'inscrit dans l'application pour offrir ses services. Il peut gérer son profil, accepter ou supprimer des demandes de rendez-vous,et signaler des publications .

Administrateur

est la personne responsable de la gestion de l'application, il gère les inscriptions des vétérinaires, supervise les alertes émises par les utilisateurs, et traite les signalements des publications.

2.3.4. Diagramme de Contexte

Le diagramme de contexte est un outil de modélisation qui représente graphiquement les interactions entre le système et son environnement extérieur. Il montre comment le système est connecté aux entités externes (acteurs système, données) avec lesquels il communique[20].

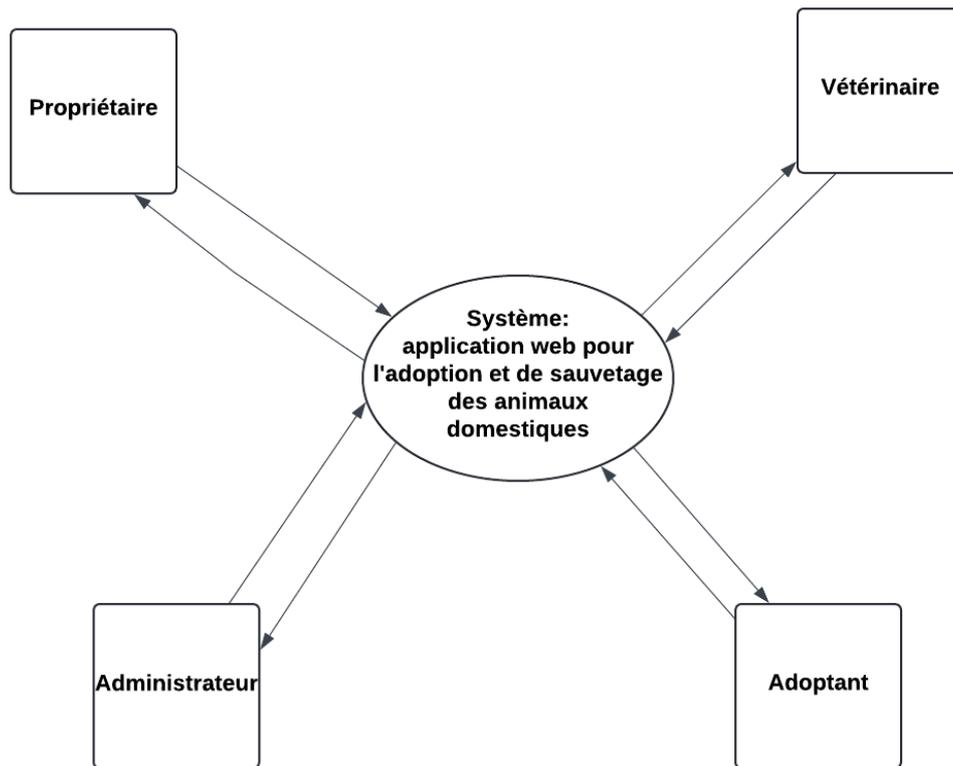


fig 2.4 : Diagramme de contexte

2.3.5. Diagramme de Cas d'Utilisation

Les cas d'utilisation, ou Use Cases en anglais, sont une méthode de modélisation intégrée à l'UML. Ils servent à décrire les interactions qui ont lieu entre un système et ses acteurs, qu'il s'agisse des utilisateurs ou d'autres systèmes [20].

Dans le premier chapitre, nous avons identifié les besoins à prendre en compte. Ainsi, nous avons défini les cas d'utilisation que notre plateforme devra gérer. Dans un diagramme de cas d'utilisation, les utilisateurs appelés acteurs (actors), sont représentés dans les différents scénarios d'utilisation.

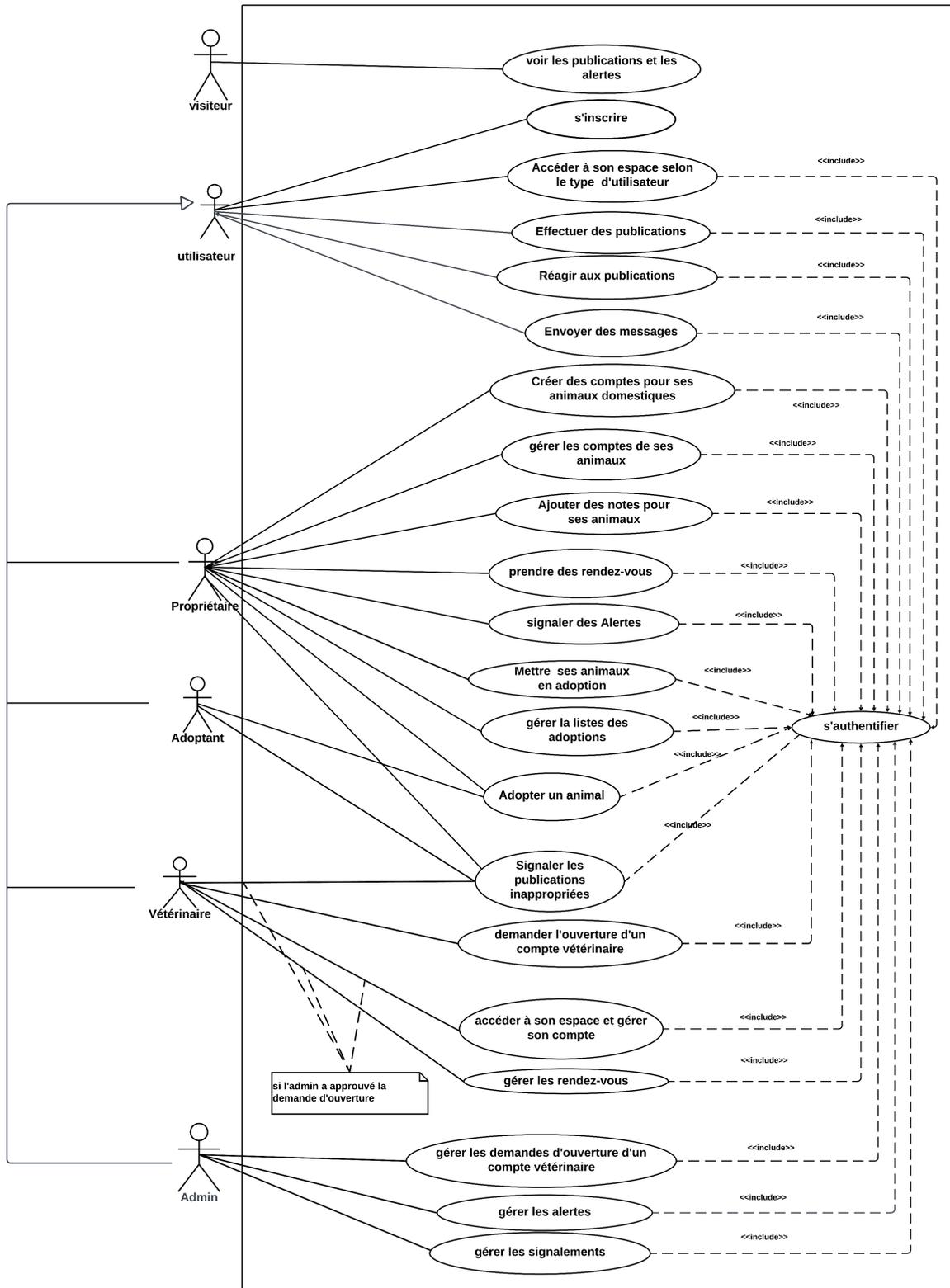


fig 2.5 : Diagramme de cas d'utilisation

2.3.6. Planification des releases

Un plan englobant plusieurs sprints est élaboré lors de la planification des releases, comme illustré par les schémas ci-dessous, en tenant compte de l'ordre de priorité :

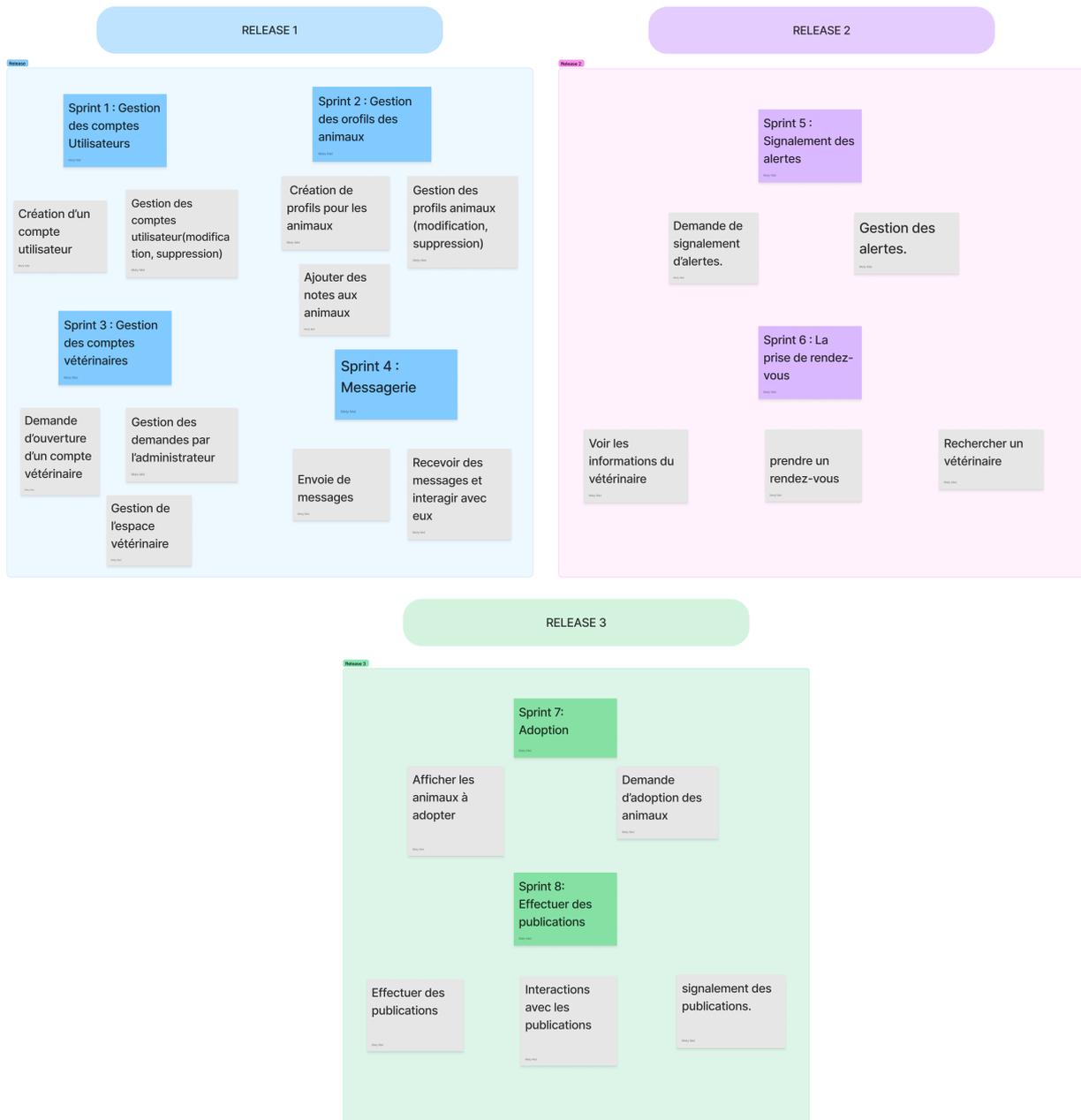


fig 2.6 : Schéma de la planification des releases

2.3.7. Backlog-Product (carnet de produit)

Chaque user-storie (histoire utilisateur) est caractérisée par une priorité définie par le Product Owner. Le Backlog de notre futur système est le suivant :

Sprint	Item	En tant que	Je veux	Priorité
1. Gestion des comptes utilisateurs	Création des comptes utilisateurs	Utilisateur	Créer un compte dans la plateforme pour que je puisse accéder à mon espace.	1

CHAPITRE 2. MÉTHODOLOGIE DE LA CONCEPTION ET SPÉCIFICATION DES BESOINS

	Gestion des comptes utilisateurs (modification, suppression)	Utilisateur	Gérer mon propre compte	2
2. Gestion des comptes des animaux	Authentification et Création de comptes pour les animaux	Propriétaire	M'authentifier pour que je puisse créer les comptes de mes animaux domestiques	1
	Gestion des comptes animaux (modification, suppression)	Propriétaire	Gérer les comptes de mes animaux	2
	Ajouter les notes aux animaux	Propriétaire	Ajouter les notes pour mes animaux pour suivre leur état	3
3. Gestion des comptes vétérinaires	Authentification	Vétérinaire	M'authentifier	1
	Demande d'ouverture d'un compte vétérinaire	Vétérinaire	Demander l'accès à mon espace pour pouvoir exercer mes rôles	2
	Gestion des demandes d'accès	Administrateur	Gérer les demandes d'accès des vétérinaires	3
	Gestion des comptes de vétérinaires après l'approbation	Vétérinaire	Gérer mon compte si l'admin a approuvé ma demande	4
4. Messagerie	Envoi de messages	Utilisateur	Envoyer des messages aux autres utilisateurs	1
	Recevoir et répondre aux messages	Utilisateur	Répondre aux messages	2
5. Signalement des alertes	Envoi de l'alerte	Propriétaire	Passer des alertes en cas d'urgence	1
	Gestion des alertes	Administrateur	Gérer les alertes (approuver ou supprimer)	2
6. Prise des rendez-vous	Voir les informations des vétérinaires	Propriétaire	Rechercher un vétérinaire par localisation	1
	Rechercher un vétérinaire	Propriétaire	Visiter le profil du vétérinaire pour connaître ses informations nécessaires	2
	Prendre un rendez-vous	Propriétaire	Prendre un rendez-vous chez un vétérinaire	3
	Gestion des rendez-vous	Vétérinaire	Gérer les rendez-vous (approuver ou supprimer)	4
7. Adoption	Afficher les animaux à adopter	Propriétaire	Afficher les animaux à adopter	1

	Demande d'adoption des animaux	Adoptant et propriétaire	Demander l'adoption des animaux	2
8. Effectuer des publications	Effectuer des publications liées aux sauvetages, adoption des animaux et interagir avec eux	Utilisateur	Créer des publications sur le sauvetage d'un animal	1
	Signalement	vétérinaire propriétaire adoptant	Signaler la publication si son contenu est hors sujet	2
	gérer les signalements	Admin	gérer tous le signalement de publication	3

TABLE 2.4 – Le Backlog de notre futur système

2.4. Charte graphique

La charte graphique peut être définie comme étant l'identité visuelle d'une application, d'une entreprise ou d'une marque, c'est un guide qui détermine tous les éléments graphiques (logo, couleur, police, typographie et maquette), leurs utilisations et leurs caractéristiques. Elle permet de donner une cohérence afin de donner à l'utilisateur des repères et lui faciliter sa visite [21].

2.4.1. Logotype

Un logo est une représentation graphique qui a pour but d'identifier de manière unique et directe une marque, une entreprise ou une application. La Figure suivante représente l'apparence du logo de notre application, qui reflète un symbole d'une patte. Ce symbole représente les animaux domestiques [22].



fig 2.7 : Logo de l'application

2.4.2. Couleurs dominantes

Notre choix est porté sur trois couleurs pour attirer l'attention de l'utilisateur tout en procurant une sensation de douceur. Les nuances de marron choisies sont inspirées par les couleurs fréquemment observées chez les animaux domestiques.

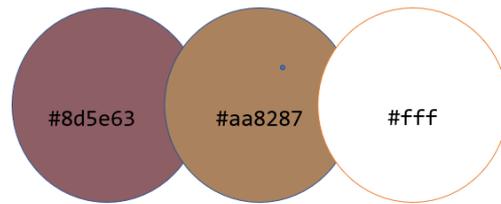


fig 2.8 : Palette de couleur

2.5. Conclusion

Dans ce chapitre, nous avons présenté un aperçu de la méthode Scrum, ainsi que ses différents composants. Nous avons également mentionné quelques notions du langage de modélisation UML. Ensuite, nous avons spécifié les besoins et les exigences de notre système à travers le diagramme de contexte et de cas d'utilisation.

Cette phase nous a permis de nous familiariser avec la méthode Scrum et de décrire de manière globale les besoins de nos utilisateurs dans le backlog de produit, ainsi que le fonctionnement désiré du système à développer afin d'en faciliter la réalisation et la maintenance.

Le chapitre suivant concerne le premier release où nous entamerons une phase très importante dans laquelle nous décrirons de manière détaillée comment ces besoins seront réalisés dans notre système.

Chapitre 3

Premier Release

3.1. Introduction

Dans ce chapitre, nous présentons le travail effectué lors de la première release, qui se compose de quatre sprints réalisés successivement :

- **Sprint 1** : Gestion des comptes utilisateurs.
- **Sprint 2** : Gestion des profils des animaux.
- **Sprint 3** : Gestion des comptes vétérinaire.
- **Sprint 4** : Messagerie.

Pour chacun de ces sprints, nous allons élaborer des diagrammes de cas d'utilisation, des descriptions textuelles et des diagrammes d'interaction pour illustrer les différentes fonctionnalités et interactions du système. Ces artefacts fournissent une compréhension détaillée du fonctionnement de l'application et servent de référence lors du développement et des tests.

3.2. Sprint 1 : Gestion des comptes utilisateurs

Ce sprint s'est étalé sur une semaine et comprend deux items comme suit :

- **Inscription et création des comptes utilisateurs** : L'utilisateur s'inscrit sur la plateforme selon le rôle qui lui correspond, notamment : un propriétaire ou un administrateur.
- **Gestion des comptes utilisateurs** : Les utilisateurs peuvent modifier leurs informations ou supprimer leurs comptes s'ils le souhaitent.

3.2.1. Diagramme de cas d'utilisation du sprint 1

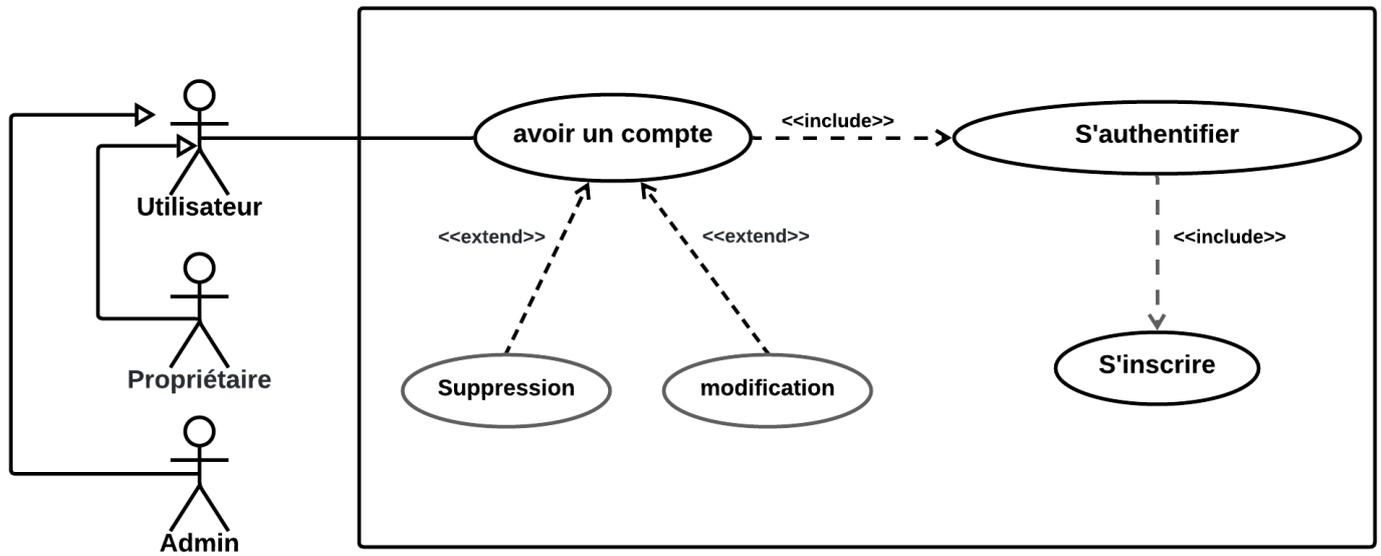


fig 3.1 : Diagramme de cas d'utilisation de sprint 1

3.2.2. Cas d'utilisation « Inscription »

Description textuelle du cas d'utilisation « Inscription »

CU	Inscription
Objectif	L'utilisateur souhaite créer un compte sur le site web pour accéder aux services ou fonctionnalités réservés aux membres inscrits.
Acteur principal	L'utilisateur
Acteur secondaire	/
Pré-condition	L'utilisateur doit avoir une adresse email .
Scénario nominal	<ol style="list-style-type: none"> 1. Accéder à la page d'accueil du site web. 2. Cliquer sur le bouton « s'inscrire ». 3. Remplir le formulaire et soumettre son inscription. 4. Choisir son type d'utilisateur. 5. Le système valide les informations et crée le compte. 6. Le compte est créé et le propriétaire peut le consulter.
Postcondition	L'utilisateur a un compte actif sur le site. L'utilisateur peut se connecter sur le site web en utilisant ses informations de connexion.
Enchainements alternatifs	Le système affiche un message d'erreur si le nombre de caractères du mot de passe est inférieur à 8 caractères.

TABLE 3.1 – Description du cas d'utilisation pour l'inscription

diagramme d'interaction de cas d'utilisation «s'inscrire»

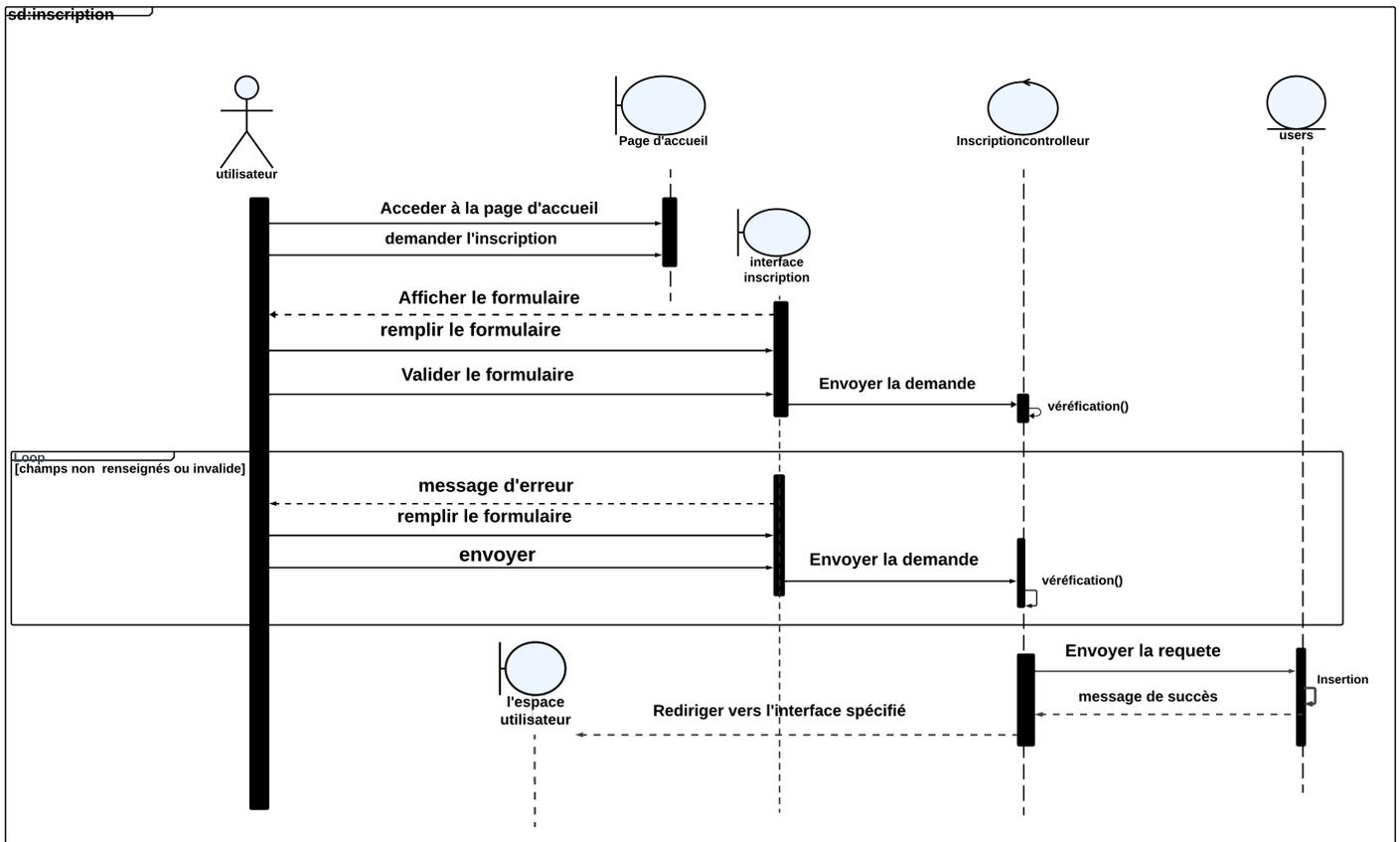


fig 3.2 : Diagramme d'interaction de cas d'utilisation « Inscription »

3.2.3. Cas d'utilisation « Se connecter »**Description textuelle du cas d'utilisation « Se connecter »**

CU	Se connecter
Objectif	L'utilisateur souhaite se connecter au site web pour accéder aux services ou fonctionnalités réservés aux membres inscrits.
Acteur principal	L'utilisateur
Acteur secondaire	/
Pré-condition	L'utilisateur doit être inscrit sur le site.
Scénario nominal	<ol style="list-style-type: none">1. Accéder à l'espace de connexion.2. Saisir son email et son mot de passe.3. Cliquer sur le bouton « Se connecter » pour valider ses informations d'identification.
Postcondition	Si les informations sont valides, l'utilisateur est connecté et dirigé vers son tableau de bord. Sinon, l'utilisateur reste sur l'interface de connexion.
Enchainements alternatifs	Le système affiche un message d'erreur si l'utilisateur a mal saisi son email ou son mot de passe.

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

Diagramme d'interaction de cas d'utilisation « Se connecter »

La figure suivante représente le diagramme d'interaction de cas d'utilisation « Se connecter ».

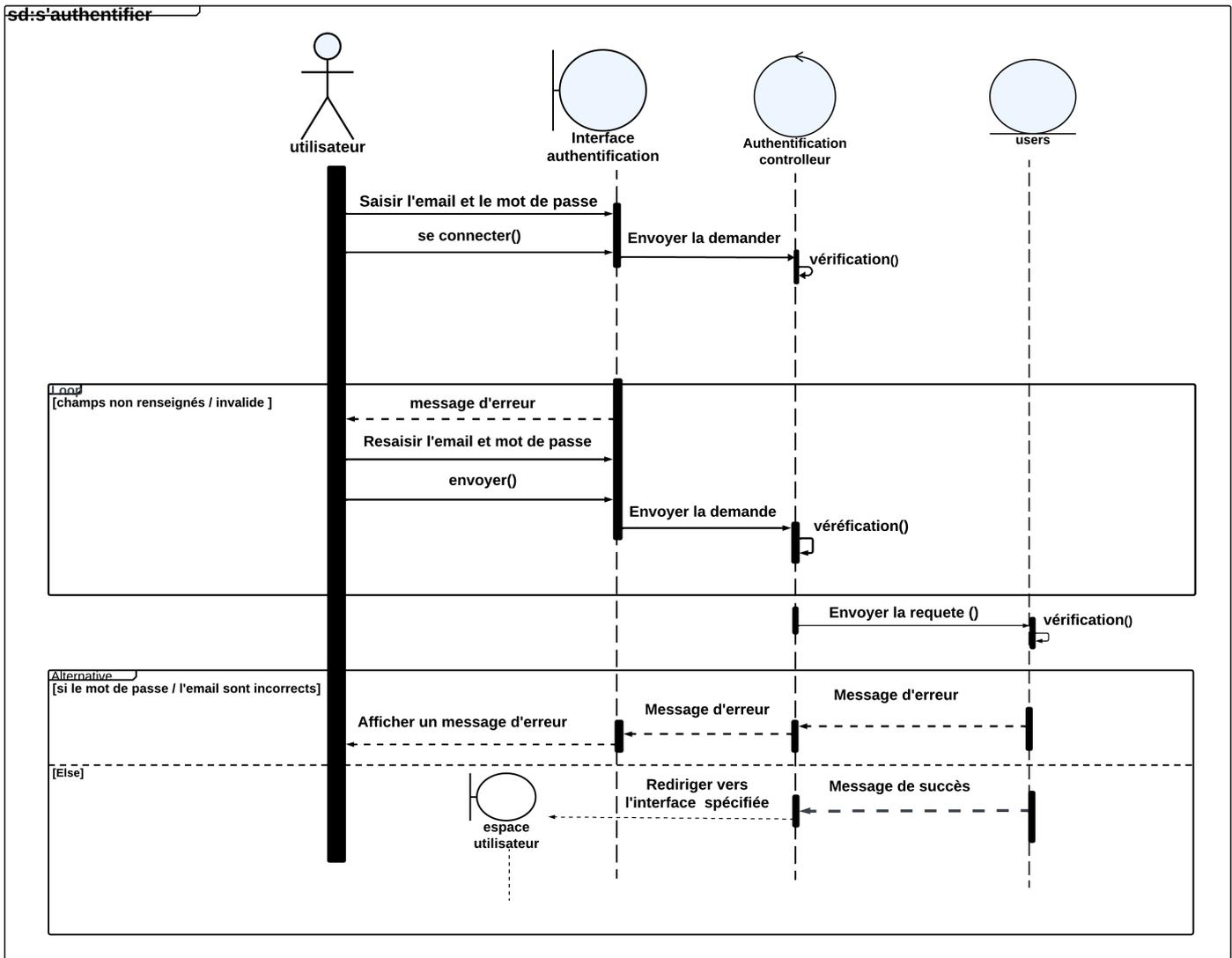


fig 3.3 : Diagramme d'interaction de cas d'utilisation « se connecter »

3.2.4. diagramme de classe de sprint01

Cette figure représente diagramme de classe de sprint 01 :

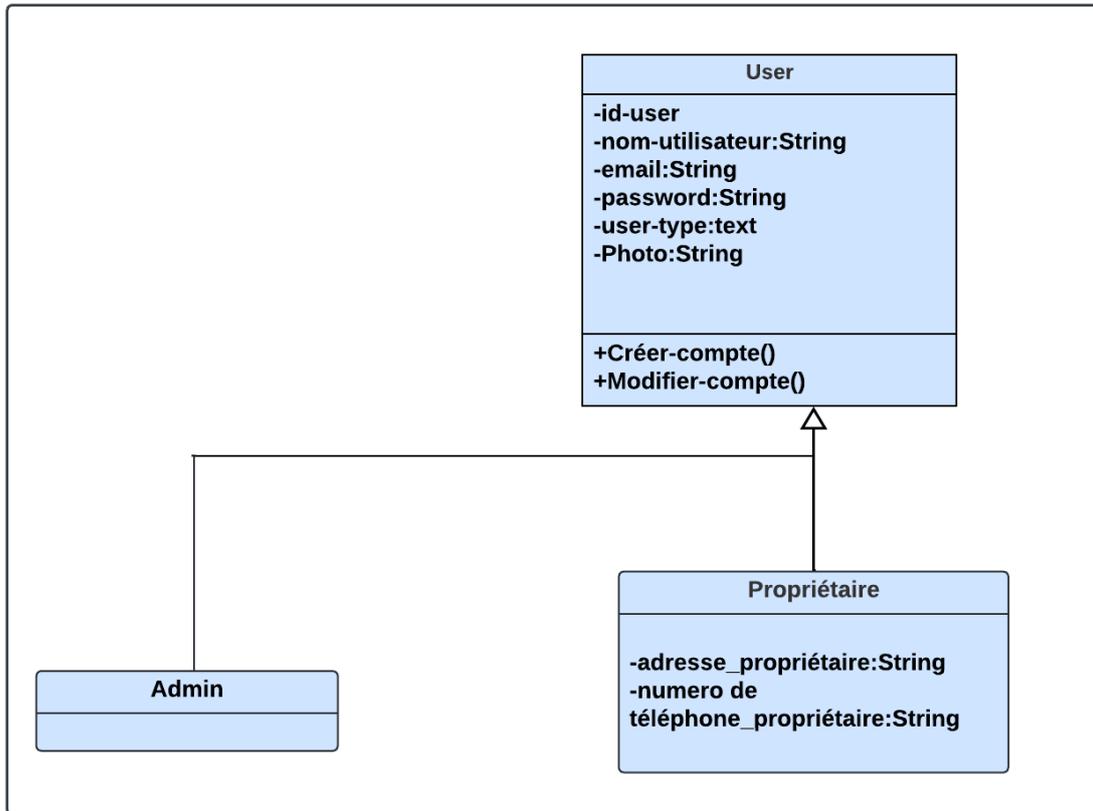


fig 3.4 : Diagramme de classe« Sprint 1»

3.3. sprint 2 : Gestion des profils des animaux

Ce sprint s'est étalé sur 12 jours et comprend trois items qui sont :

- **création des profils des animaux** : Le propriétaire crée des profils pour ses animaux.
- **Gestion de profil animal** : Les propriétaires peuvent modifier et supprimer les informations de leurs animaux domestiques, assurant ainsi la mise à jour des données.
- **Ajout de note pour chaque animal** : Le propriétaire peut ajouter des notes pour chacun de ses animaux, cela lui permet une gestion complète et organisée de tous les aspects de la vie d'un animal tels que le suivi de la santé et des activités de l'animal, la gestion de son alimentation, etc.

3.3.1. Diagramme de cas d'utilisation du sprint 2

Le diagramme de cas d'utilisation suivant illustre de manière générale les différents cas d'utilisation correspondant aux items du deuxième sprint.

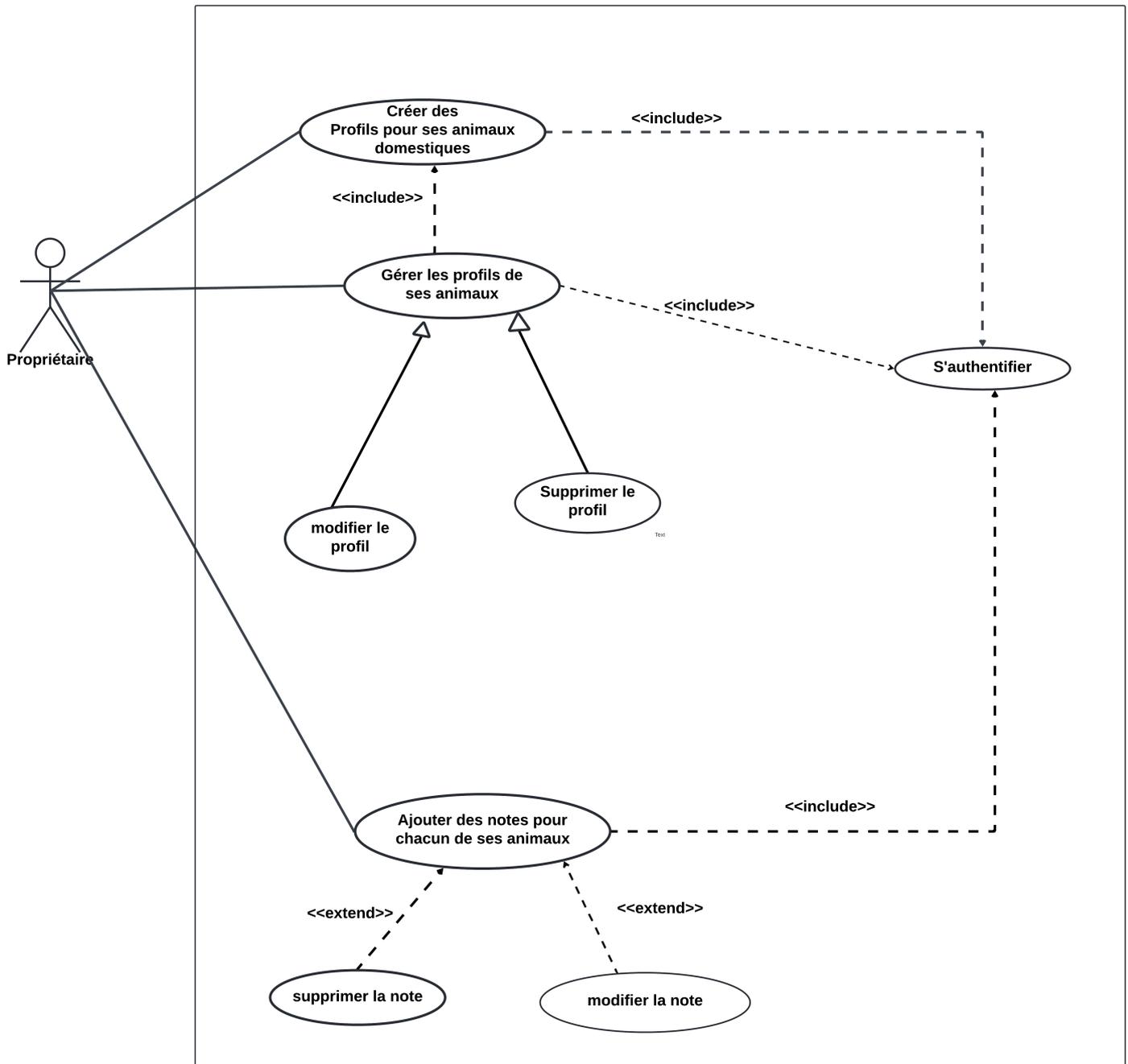


fig 3.5 : Diagramme de cas d'utilisation « sprint2 »

3.3.2. Cas d'utilisation « Créer des Profils pour les animaux domestiques »

Description textuelle du cas d'utilisation « Créer des comptes pour les animaux domestiques »

CU	Création d'un profil pour l'animal
But	Faciliter la gestion de suivi des informations et des soins nécessaires à leur bien-être
Acteur principal	Propriétaire de l'animal
Acteur secondaire	/
Pré-condition	Le propriétaire de l'animal doit être inscrit sur le site

Scénario nominal	<ol style="list-style-type: none">1. Le propriétaire sélectionne l'option « créer Profil » dans l'application.2. Le système affiche un formulaire .3. Le propriétaire saisit les informations requises sur l'animal (nom, espèce, âge, etc.).4. Le système valide les informations et crée le Profil.5. Le Profil est créé et le propriétaire peut le consulter.
Post-condition	Un nouveau Profil de l'animal a été créé et le propriétaire peut y accéder
Enchaînements alternatifs	Si les informations sur l'animal sont incomplètes ou incorrectes, le système affiche un message d'erreur

TABLE 3.3 – Description textuelle du cas d'utilisation « Créer des comptes pour les animaux domestiques »

Diagramme d'interaction de cas d'utilisation « Créer des Profils pour les animaux domestiques »

La figure suivante représente le diagramme d'interaction de cas d'utilisation « Créer des Profils pour les animaux domestiques ».

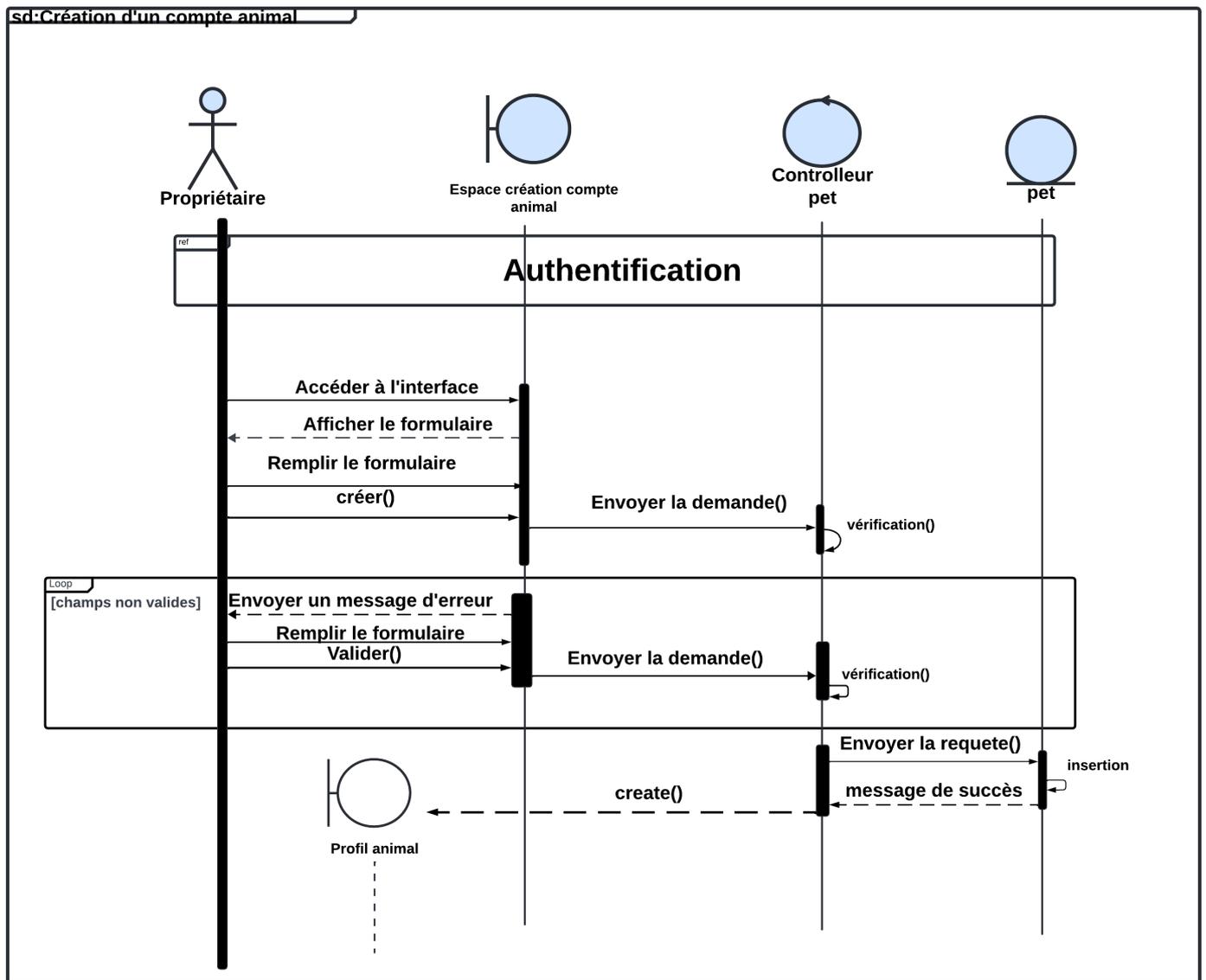


fig 3.6 : Diagramme d'interaction de cas d'utilisation « créer Profil animal »

3.3.3. Cas d'utilisation « Gérer les "Profils des animaux domestiques" »

Description textuelle du cas d'utilisation « Gérer les "Profils des animaux domestiques" »

CU	Gérer les Profils des animaux domestiques
But	Centraliser, simplifier la gestion des informations relatives aux animaux
Acteur principal	Propriétaire
Acteur secondaire	/
Pré-condition	L'utilisateur doit être connecté et avoir au moins un Profil d'animal

Scénario nominal	<ol style="list-style-type: none"> 1. Le propriétaire se connecte à son compte sur l'application. 2. Le propriétaire sélectionne l'animal dont il souhaite gérer le Profil. 3. Le propriétaire peut effectuer les actions suivantes : <ul style="list-style-type: none"> — Modifier les informations de l'animal : mise à jour des données de l'animal. — Supprimer le Profil de l'animal : en cas de besoin, l'utilisateur peut supprimer le compte associé à son animal.
Post-condition	Les informations du Profil animal sont mises à jour au niveau de la base de données. En cas de suppression, le compte de l'animal sera retiré de la base de données
Enchaînements alternatifs	Si les modifications sur l'animal sont incomplètes ou incorrectes, le système affiche un message d'erreur

TABLE 3.4 – la description de cas d'utilisation Gérer les Profils des animaux

Diagramme d'interaction de cas d'utilisation « Gérer les "Profils des animaux domestiques" »

La figure suivante présente le diagramme d'interaction de cas d'utilisation « Modification de profil de l'animal » :

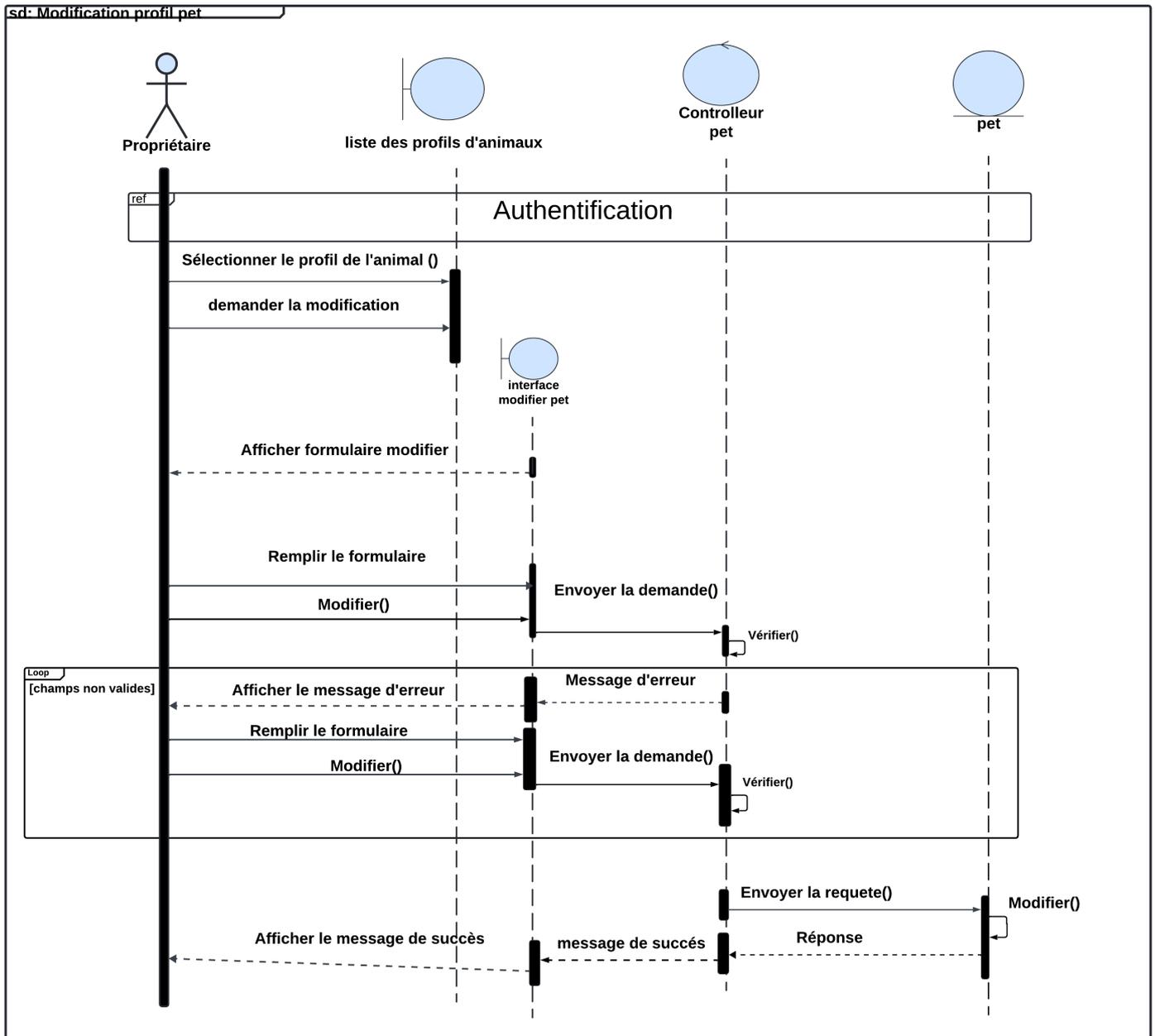


fig 3.7 : Diagramme d'interaction de cas d'utilisation « modifier compte animal »

Diagramme d'interaction de cas d'utilisation « Supprimer un animal »

La figure suivante présente le diagramme d'interaction de cas d'utilisation « Supprimer un animal » :

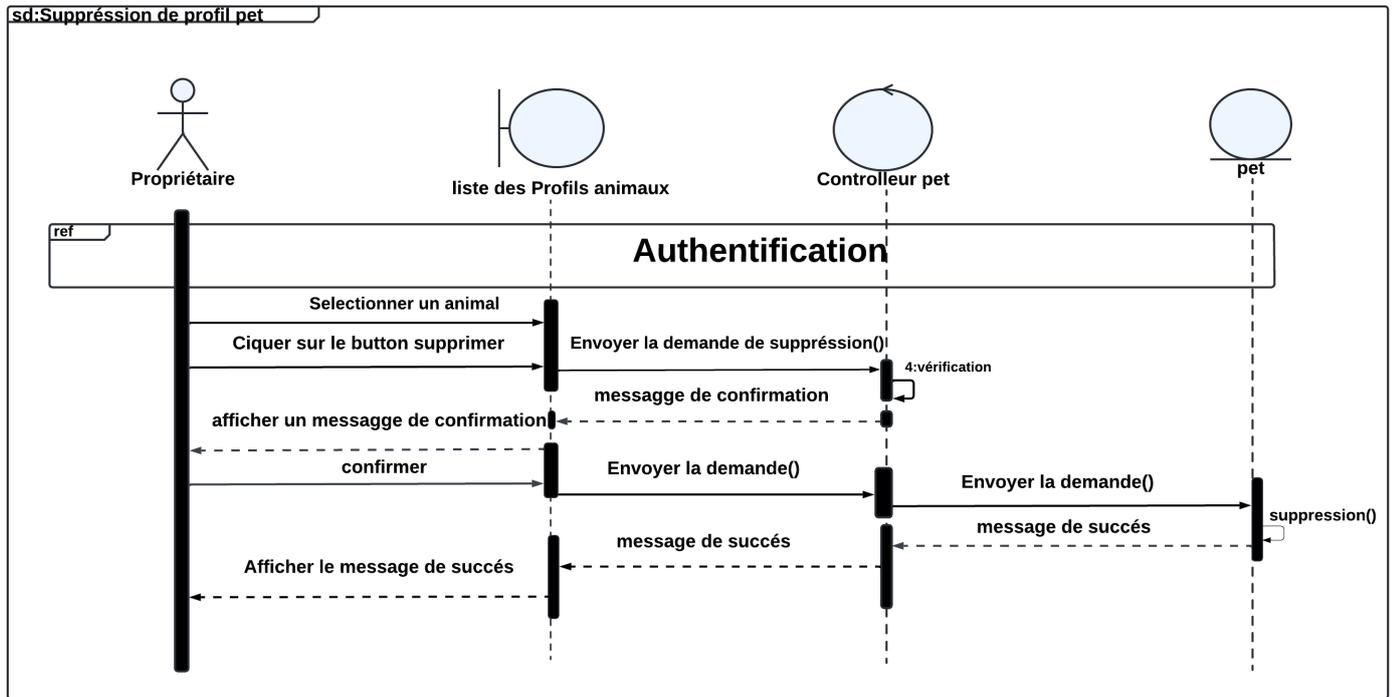


fig 3.8 : Diagramme d'interaction de cas d'utilisation «Supprimer animal »

3.3.4. Cas d'utilisation « Ajouter une note pour un animal »

Description textuelle du cas d'utilisation « Ajouter une note pour un animal »

CU	Ajouter une note pour un animal
But	Faciliter la gestion de suivi des informations et des soins nécessaires à leur bien-être
Acteur principal	Propriétaire
Acteur secondaire	/
Pré-condition	L'utilisateur doit être connecté et avoir les droits nécessaires pour ajouter des notes au profil de l'animal
Scénario nominal	<ol style="list-style-type: none"> 1. Le propriétaire clique sur l'option « Ajouter une note ». 2. Le propriétaire sélectionne l'animal concerné. 3. Le système présente un formulaire pour saisir le contenu de la note. 4. Le propriétaire entre le détail de la note (date, description, etc.). 5. Le propriétaire soumet la note. 6. Le système enregistre la note dans le dossier de l'animal et confirme l'ajout au propriétaire.
Post-condition	Une note est ajoutée au compte de l'animal accessible pour consultation ou mise à jour ultérieure

Enchaînements alternatifs	Si les informations sont incomplètes ou incorrectes, le système affiche un message d'erreur
----------------------------------	---

TABLE 3.5 – Description textuelle du cas d'utilisation « Ajouter une note pour un animal »

Diagramme d'interaction de cas d'utilisation « Ajouter une note pour un animal »

La figure suivante présente le diagramme d'interaction de cas d'utilisation « Ajouter une note pour un animal ».

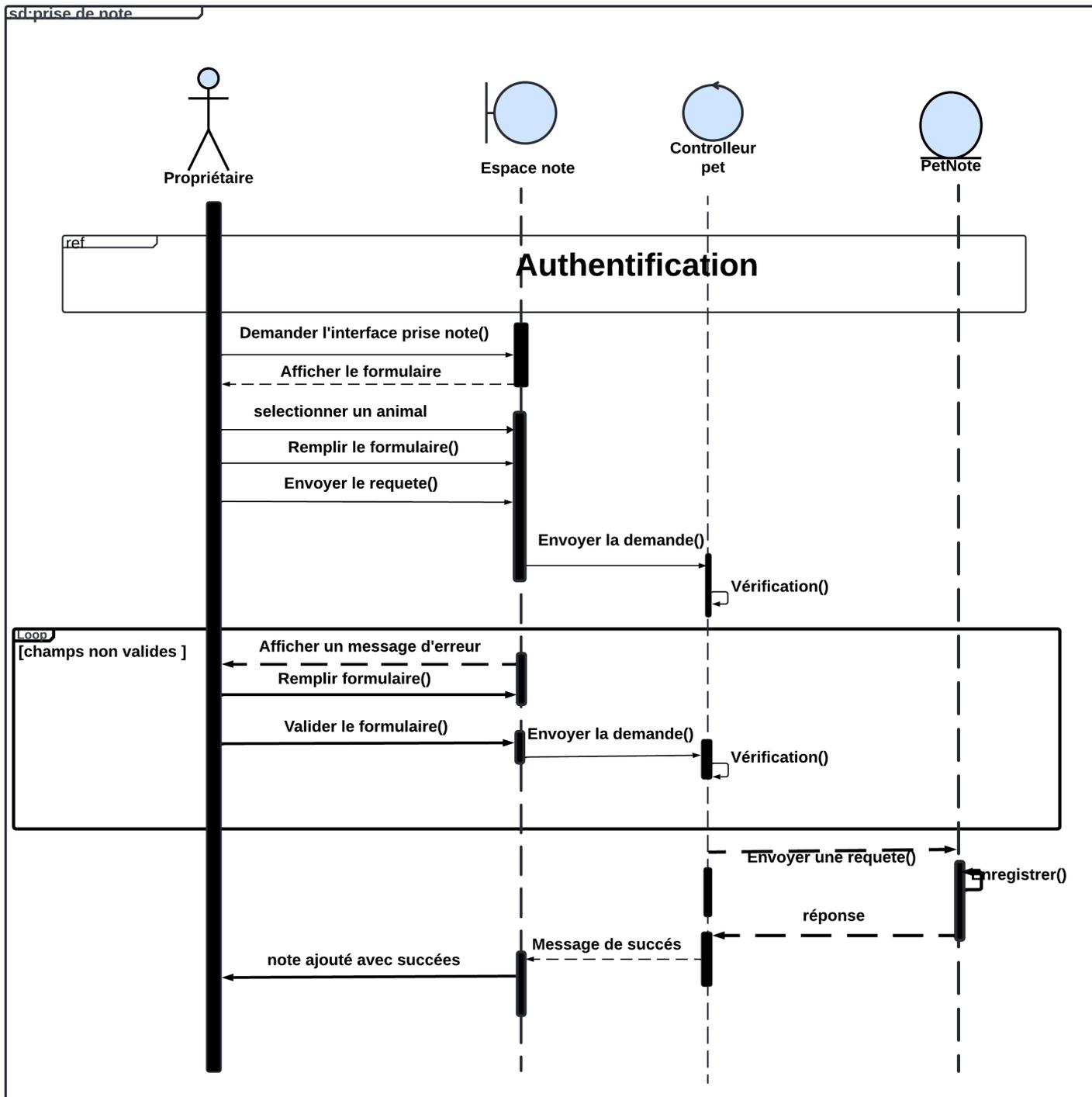


fig 3.9 : Diagramme d'interaction de cas d'utilisation « Ajouter une note pour un animal»

Diagramme de classe d'utilisation du deuxième sprint

La figure ci-dessous représente le diagramme de classe pour le sprint 2 « Compte animal»

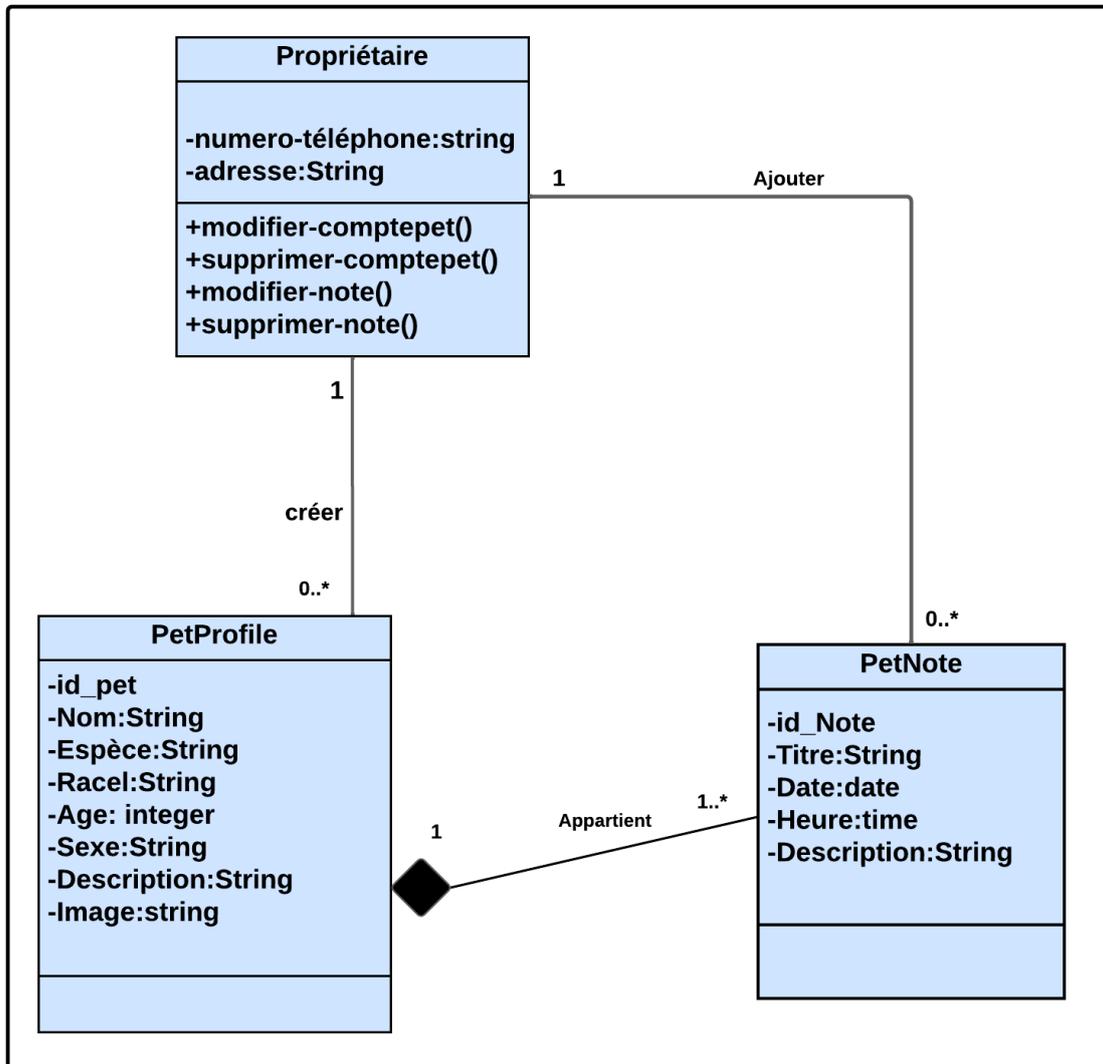


fig 3.10 : Diagramme de classe du sprint02»

3.4. Sprint 3 : Gestion des demandes d'accès du vétérinaire

3.4.1. Diagramme de cas d'utilisation du troisième sprint

Le diagramme de cas d'utilisation suivant représente de manière générale les cas d'utilisation constituant les items du troisième sprint :

3.4.2. Cas d'utilisation « Demande d'ouverture d'un compte vétérinaire »

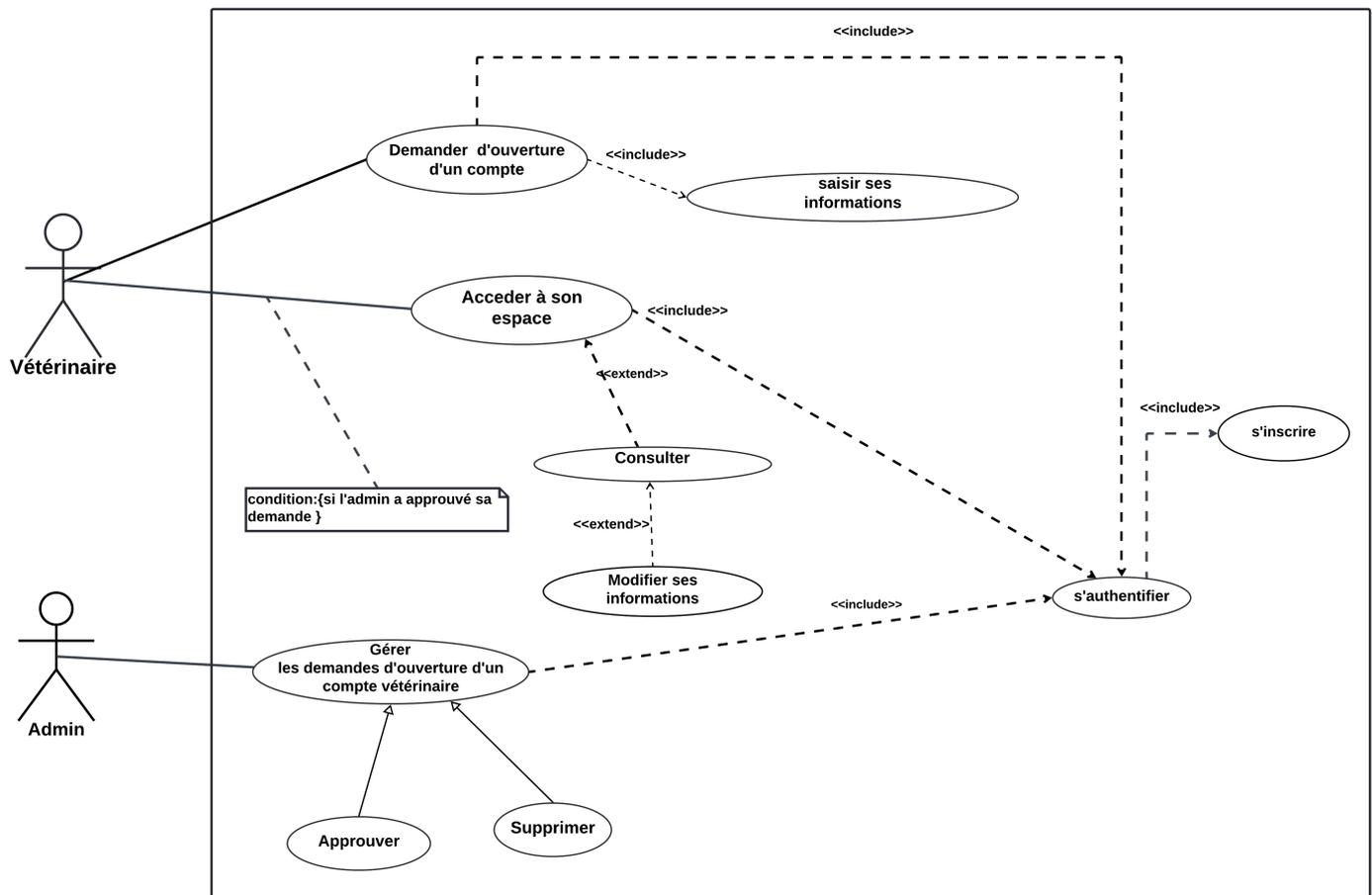


fig 3.11 : Diagramme de cas d'utilisation « Demande d'ouverture d'un compte vétérinaire »

Description textuelle du cas d'utilisation « Demande d'ouverture d'un compte vétérinaire »

CU	vétérinaire
But	Ce cas traite les demandes d'ouverture d'un compte professionnel « vétérinaire » afin qu'il puisse accéder à son espace
Acteur principal	Vétérinaire
Acteur secondaire	/
Pré-condition	Le vétérinaire doit être authentifié sur la plateforme en tant qu'un simple utilisateur

Scénario nominal	<ol style="list-style-type: none"> 1. Le vétérinaire accède à la fonctionnalité de demande d'ouverture d'un compte vétérinaire. 2. Le système présente un formulaire où le vétérinaire doit saisir ses informations professionnelles (par exemple : le nom du cabinet, la localisation, etc.). 3. Le vétérinaire remplit le formulaire avec ses informations. 4. Le vétérinaire soumet la demande d'accès. 5. Le système enregistre la demande et envoie un accusé de réception au vétérinaire.
Post-condition	La demande d'accès est enregistrée dans le système et sera examinée par l'administrateur
Enchaînements alternatifs	Si les informations sont incomplètes ou incorrectes, le système affiche un message d'erreur

TABLE 3.6 – Description textuelle du cas d'utilisation « Demande d'ouverture d'un compte vétérinaire »

Diagramme d'interaction « Demande d'ouverture d'un compte vétérinaire »

La figure suivante représente le diagramme d'interaction de « Demande d'ouverture d'un compte vétérinaire » :

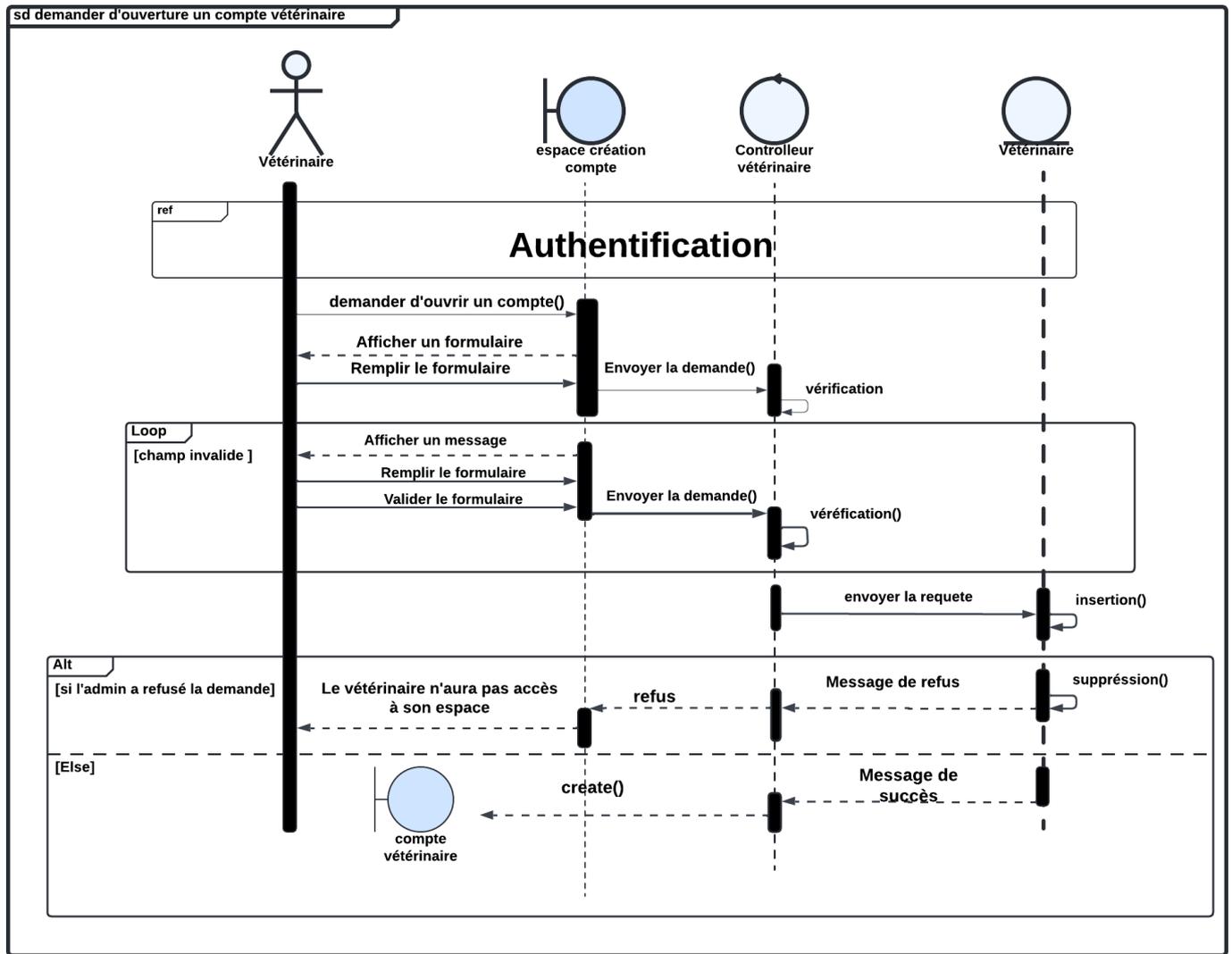


fig 3.12 : Diagramme d'interaction de « demande d'ouverture un compte vétérinaire »

3.4.3. Cas d'utilisation « Gestion des demandes d'ouverture d'un compte vétérinaire »

Description textuelle du cas d'utilisation « Gestion des demandes d'ouverture d'un compte vétérinaire »

CU	Gérer les demandes d'ouverture d'un compte vétérinaire
But	Examiner et traiter les demandes d'accès soumises par les vétérinaires pour accéder à leur espace personnel sur la plateforme
Acteur principal	Administrateur
Acteur secondaire	/
Pré-condition	<ul style="list-style-type: none"> — L'administrateur doit s'authentifier. — Les demandes d'ouverture de compte ont été mises à jour par les vétérinaires et sont en attente.

Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur accède à l'espace de gestion des vétérinaires. 2. Le système affiche la liste des demandes d'accès en attente. 3. L'administrateur sélectionne une demande pour la consulter en détail. 4. Après la vérification des informations, l'administrateur décide soit d'approuver soit de refuser la demande.
Post-condition	La demande d'accès est soit approuvée et l'accès est accordé, soit refusée et le vétérinaire ne pourra pas accéder à son espace
Enchaînements alternatifs	/

TABLE 3.7 – la description de cas d'utilisation Gestion des demandes d'ouverture d'un compte vétérinaire

Diagramme d'interaction de « Gestion des demandes d'ouverture d'un compte vétérinaire »

La figure suivante représente le diagramme d'interaction de « Gestion des demandes d'ouverture d'un compte vétérinaire » :

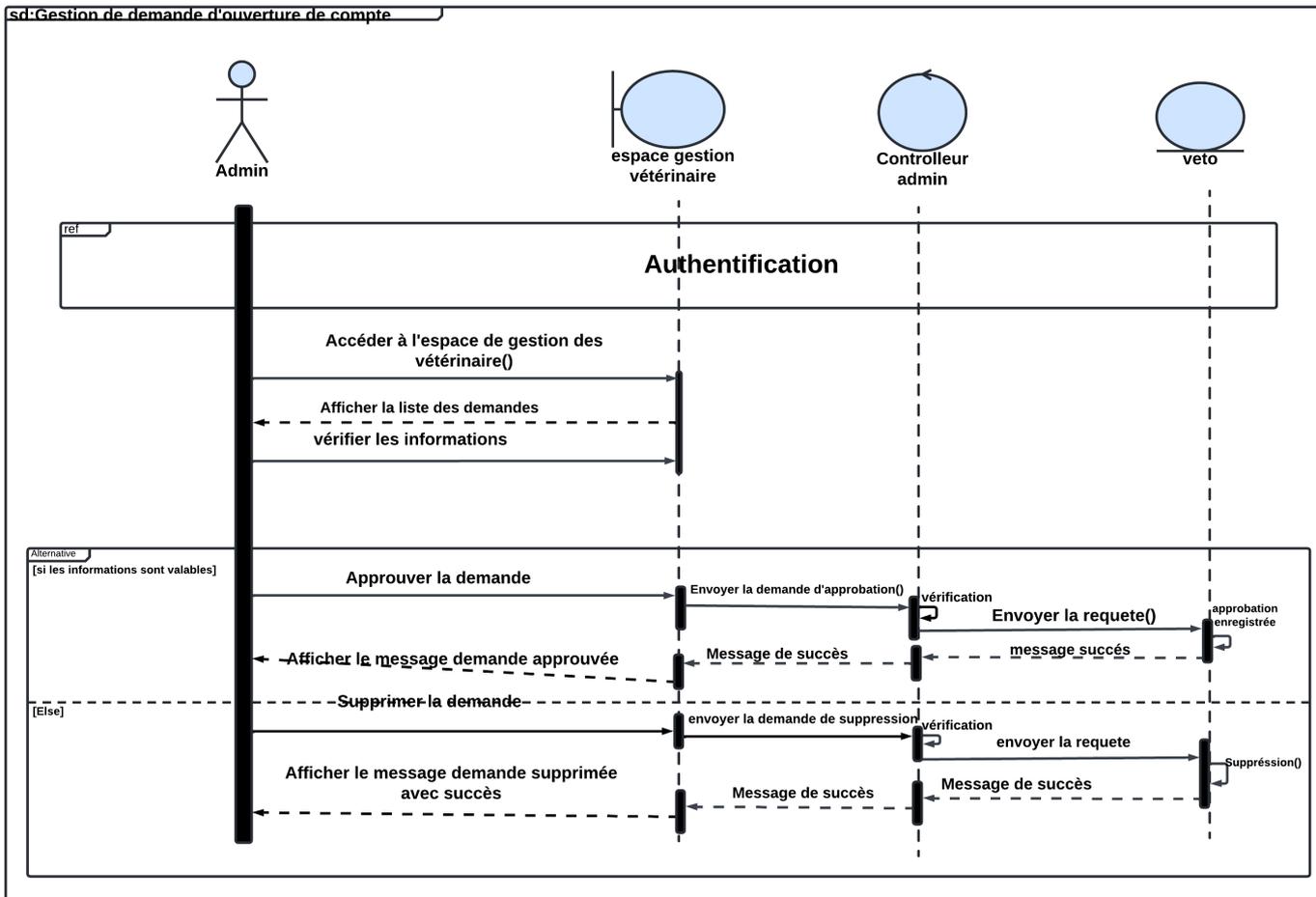


fig 3.13 : Diagramme d'interaction de « gestion des demandes d'ouverture comptes vétérinaires »

3.4.4. Cas d'utilisation « Modification d'un compte vétérinaire »

Description textuelle du cas d'utilisation « Modification d'un compte vétérinaire »

CU	Modification d'un compte vétérinaire
But	Permettre aux vétérinaires de modifier leurs informations personnelles et professionnelles
Acteur principal	Vétérinaire
Acteur secondaire	/
Pré-condition	<ul style="list-style-type: none"> — Le vétérinaire doit être authentifié. — La demande d'accès du vétérinaire doit être acceptée.

Scénario nominal	<ol style="list-style-type: none"> 1. Le vétérinaire accède à la page de son profil. 2. Le vétérinaire clique sur le bouton pour modifier ses informations. 3. Le vétérinaire modifie les champs souhaités (par exemple, nom, email, mot de passe). 4. Le vétérinaire soumet le formulaire. 5. Le système valide les données saisies. 6. Si les données sont valides, le système met à jour les informations du vétérinaire dans la base de données. 7. Le système affiche un message de confirmation de la mise à jour réussie. 8. L'utilisateur est redirigé vers la page de son profil avec les informations modifiées.
Post-condition	Les informations des profils sont mises à jour dans la base de données
Enchaînements alternatifs	Les informations de l'utilisateur ne sont pas mises à jour en raison d'erreurs de validation ou de problèmes techniques

TABLE 3.8 – la description de cas d'utilisation modification compte veterinaire

Diagramme d'interaction de cas d'utilisation modifier compte vétérinaire

La figure suivante représente le diagramme d'interaction de cas d'utilisation modifier compte vétérinaire

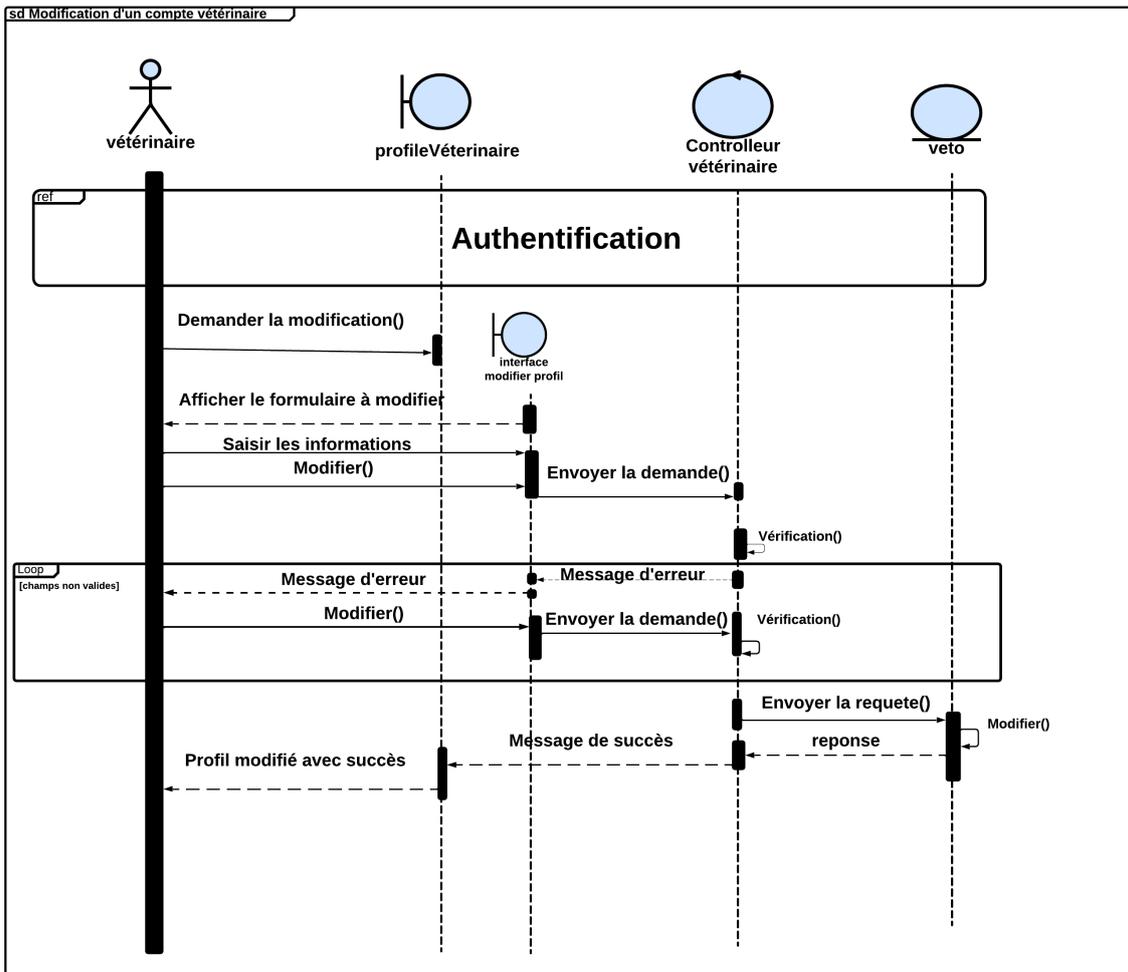


fig 3.14 : diagramme d'interaction modifier compte vétérinaire

Diagramme de classe de sprint3

La figure suivante représente le diagramme de classe du sprint 3

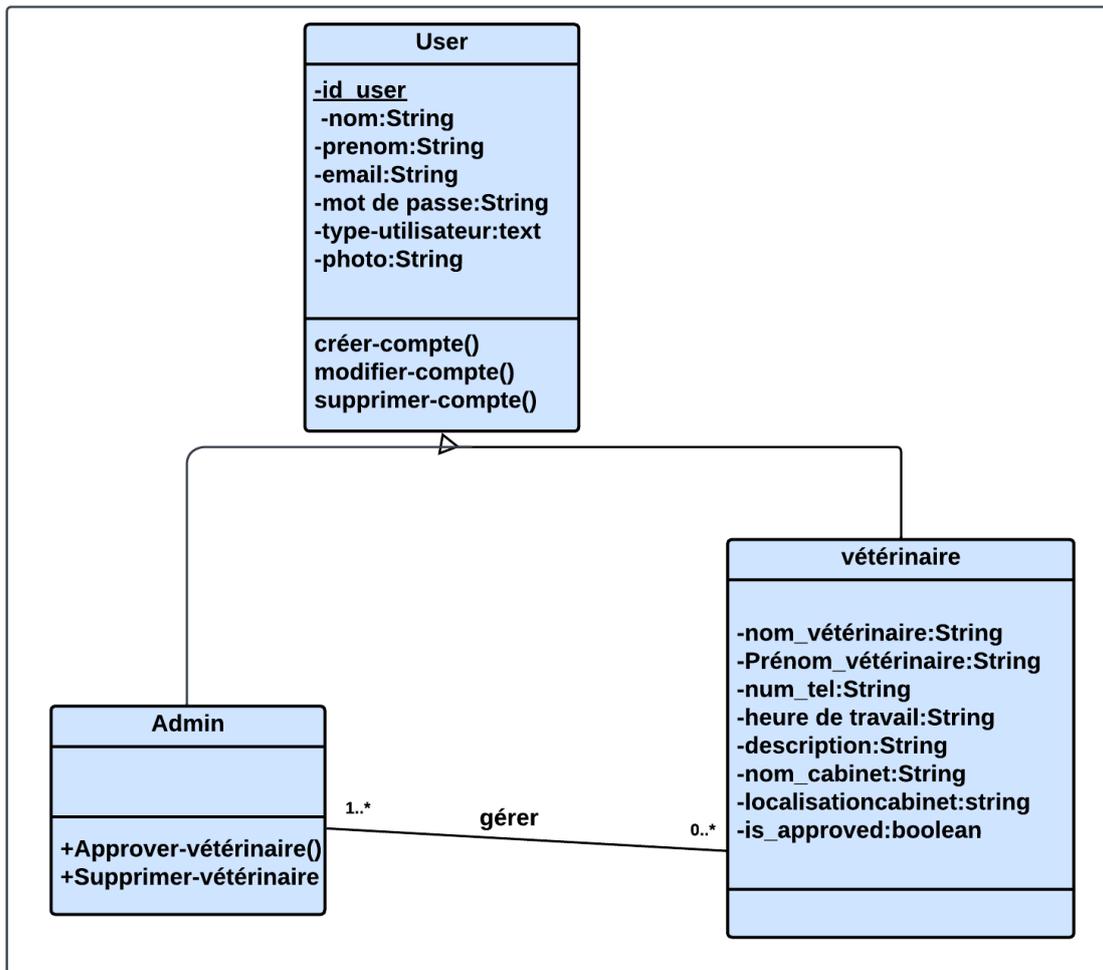


fig 3.15 : diagramme de classe sprint 3

3.5. Sprint 4 : Messagerie

3.5.1. Diagramme de cas d'utilisation du sprint 4

Le diagramme de cas d'utilisation suivant représente de manière générale les cas d'utilisation constituant les items du quatrième sprint :

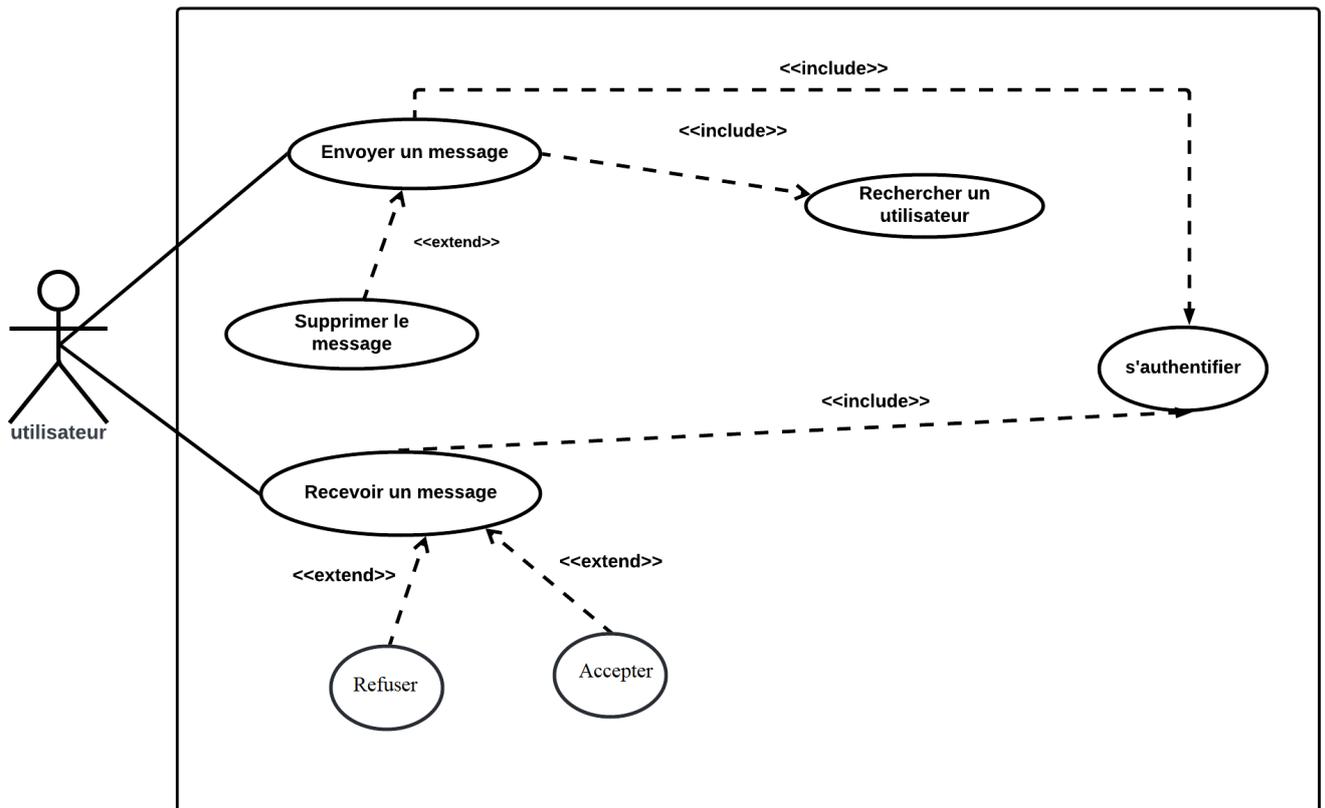


fig 3.16 : Diagramme d'interaction de cas d'utilisation « Envoyer des messages »

3.5.2. Cas d'utilisation « Envoyer des messages »

Description textuelle du cas d'utilisation « Envoyer des messages »

CU	Envoyer des messages
But	Permettre à l'utilisateur d'envoyer des messages à un autre utilisateur de l'application
Acteur principal	Utilisateur
Acteur secondaire	/
Pré-condition	<ul style="list-style-type: none"> — L'utilisateur doit s'authentifier — Le destinataire doit être disponible dans l'application
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur accède à l'espace de messagerie. 2. L'utilisateur recherche et sélectionne le destinataire du message. 3. L'utilisateur rédige son message dans le champ de texte. 4. L'utilisateur envoie le message. 5. Le destinataire reçoit le message et peut y répondre.

Post-condition	Les messages sont correctement envoyés aux destinataires. L'utilisateur et le destinataire peuvent continuer leur discussion
Enchaînements alternatifs	Si la connexion échoue, l'utilisateur ne peut pas envoyer ou recevoir des messages

TABLE 3.9 – la description de cas d'utilisation Envoyer des messages

Diagramme d'interaction de cas d'utilisation « Envoyer des messages »

La figure suivante représente le diagramme d'interaction de cas d'utilisation « Envoyer des messages » :

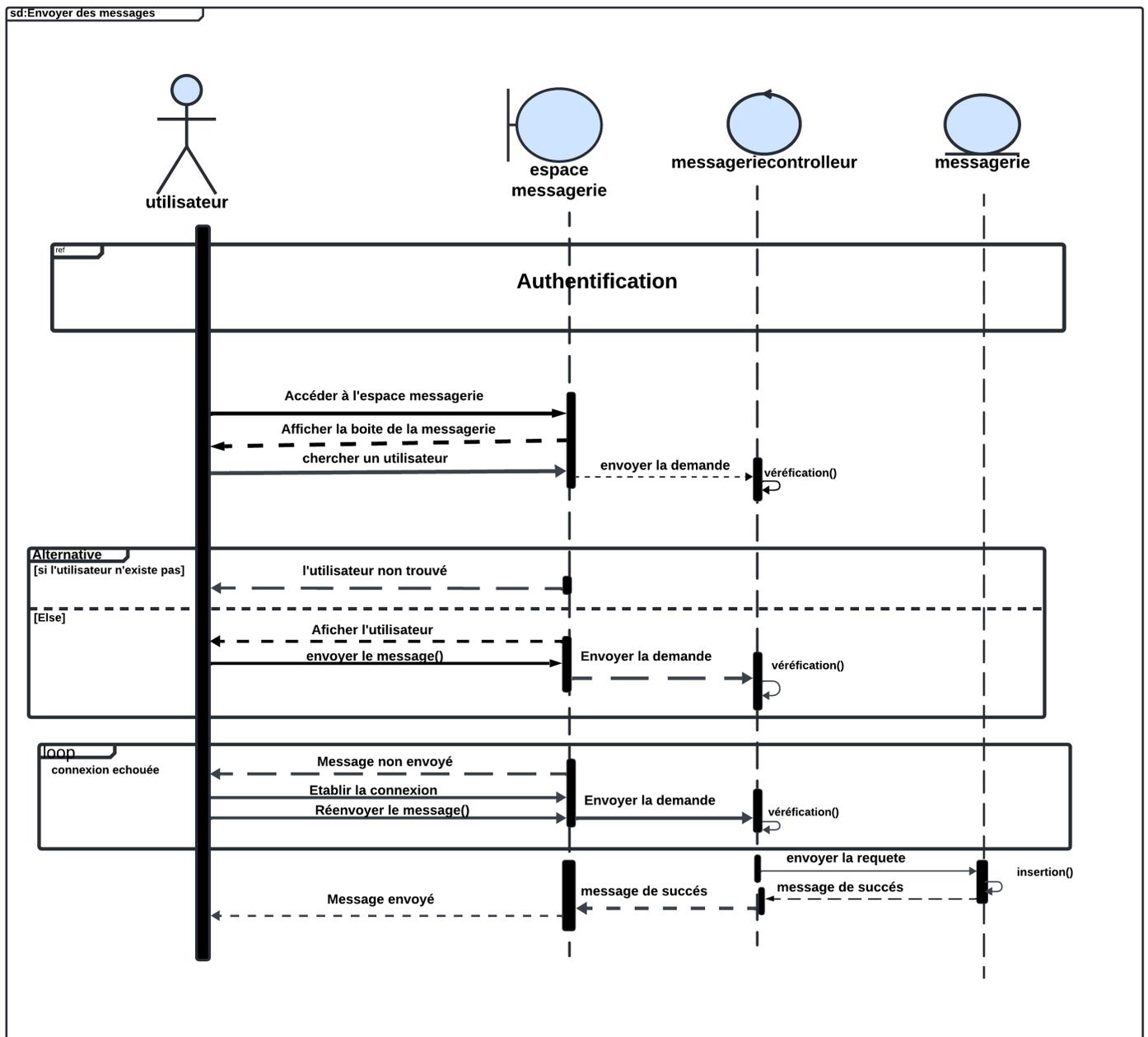


fig 3.17 : Diagramme d'interaction de cas d'utilisation « Envoyer des messages»

3.5.3. Diagramme de classe du quatrième sprint

La figure suivante représente le diagramme de classe pour le sprint4 « messagerie »

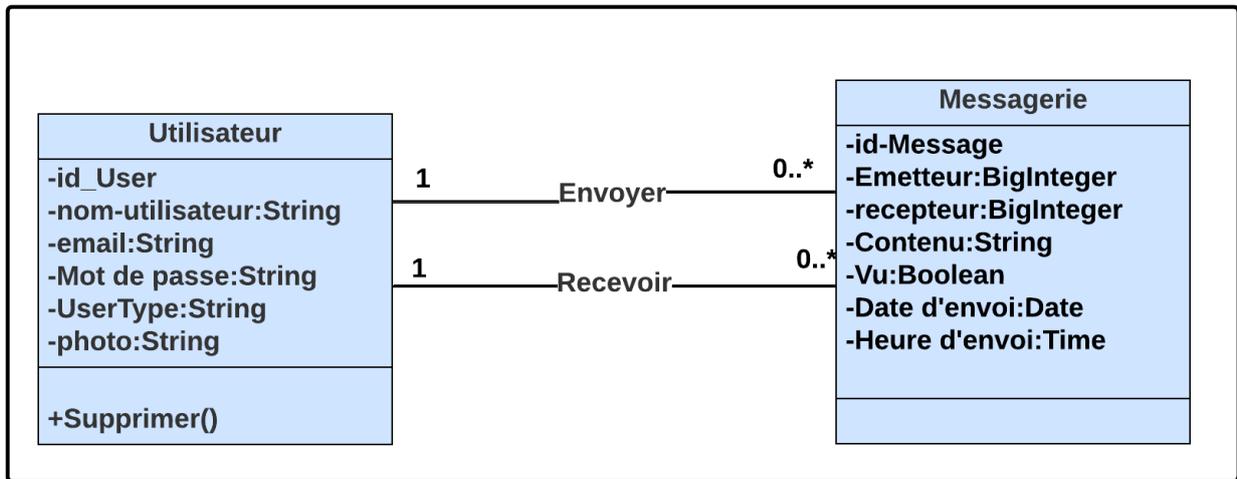


fig 3.18 : Diagramme de classe du sprint4.

3.6. Conclusion

Le premier release de notre projet a marqué un début prometteur avec trois sprints réussis. Le premier sprint a permis la création des comptes utilisateurs, offrant une base solide pour notre application. Le second sprint a introduit une fonctionnalité cruciale, la gestion des comptes pour les animaux. Le quatrième sprint concernant la messagerie via une API, a également été réalisé avec succès. Cependant le troisième sprint qui concerne la demande d'accès pour les vétérinaires, nécessite des améliorations pour fonctionner pleinement . Ces avancées posent les fondations d'un système robuste et évolutif pour l'avenir. Dans le prochain chapitre nous allons entamer le deuxième release.

Chapitre 4

deuxième release

4.1. Introduction

Dans ce chapitre, nous allons présenter le travail effectué durant le deuxième release qui comprend :

- Le sprint 5 : Signalement des alertes.
- Le sprint 6 : La prise de rendez-vous.

Pour chacun de ces sprints, nous allons élaborer des diagrammes de cas d'utilisation ainsi que les descriptions textuelles pour les cas d'utilisation et les diagrammes d'interactions.

4.2. Sprint 5 : signalement des alertes

Le cinquième sprint implémente un système permettant aux propriétaires de signaler des urgences, telles que la perte d'un animal ou la nécessité urgente d'un médicament. Les alertes sont ensuite gérées par l'administrateur, qui décide de les approuver pour les rendre visibles sur la plateforme. Ce sprint s'est étalé sur 10 jours, il contient deux items :

- Demande de signalement d'alertes.
- Gestion des alertes.

4.2.1. Diagramme de cas d'utilisation du sprint 5 :

Le diagramme de cas d'utilisation ci-dessous illustre les cas d'utilisation correspondant aux items du cinquième sprint :

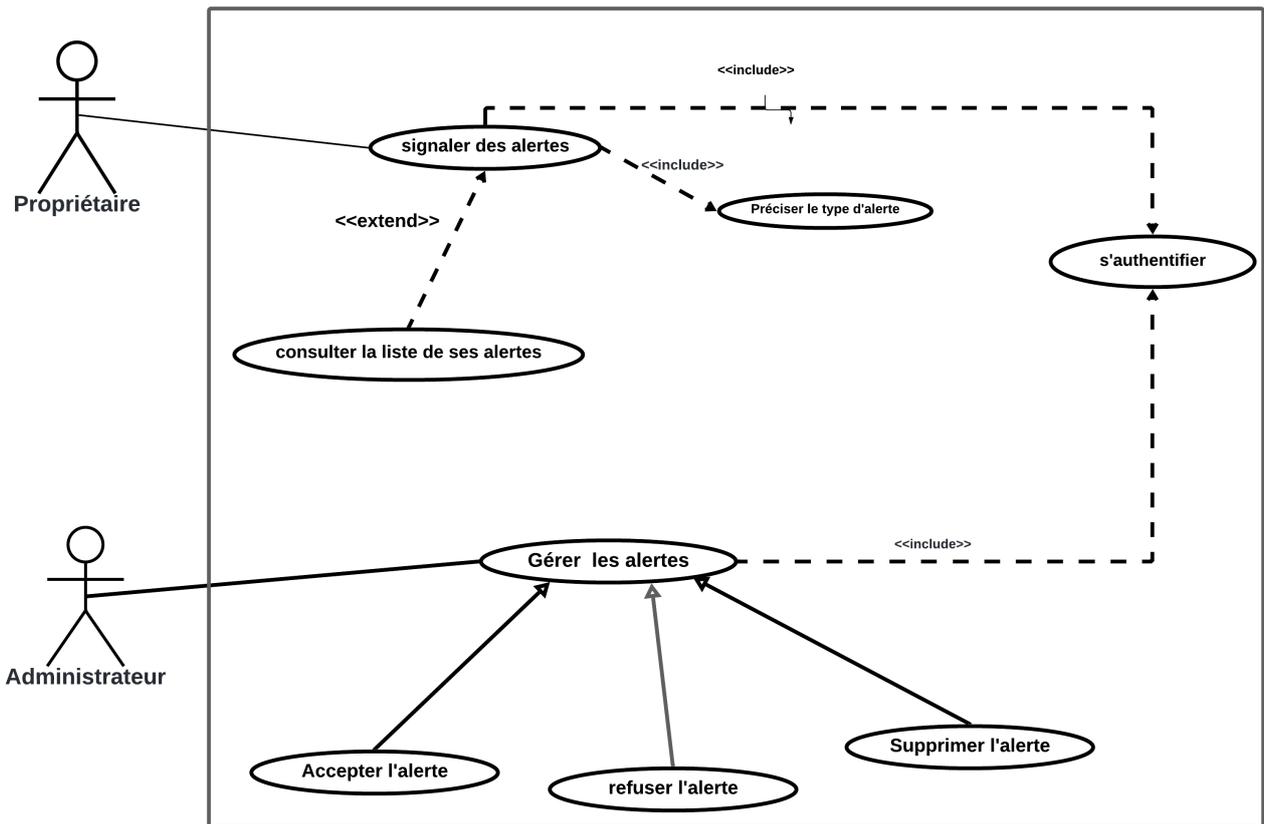


fig 4.1 : Diagramme de cas d'utilisation du sprint 5.

4.2.2. Cas d'utilisation « Signaler des alertes »

la description textuelle de Cas d'utilisation « Signaler des alertes »

CU	Signaler des alertes
Acteur Principal	Propriétaire de l'animal
Objectif	Signaler une situation d'urgence concernant un animal
Précondition	Le propriétaire doit être authentifié
Scénario Principal	<ol style="list-style-type: none"> 1. Le propriétaire accède à l'espace alerte dans l'application. 2. Le propriétaire choisit le type de l'alerte, puis la rédige. 3. Le propriétaire soumet l'alerte pour qu'elle soit traitée par l'administrateur.
Cas d'utilisation alternatifs	La demande ne sera pas envoyée si le propriétaire oublie de remplir un champ du formulaire.

TABLE 4.1 – la description de cas d'utilisation signaler alerte

Diagramme d'interaction de cas d'utilisation 'Signaler des alertes'

La figure suivante représente le diagramme d'interaction de cas d'utilisation 'signaler alerte

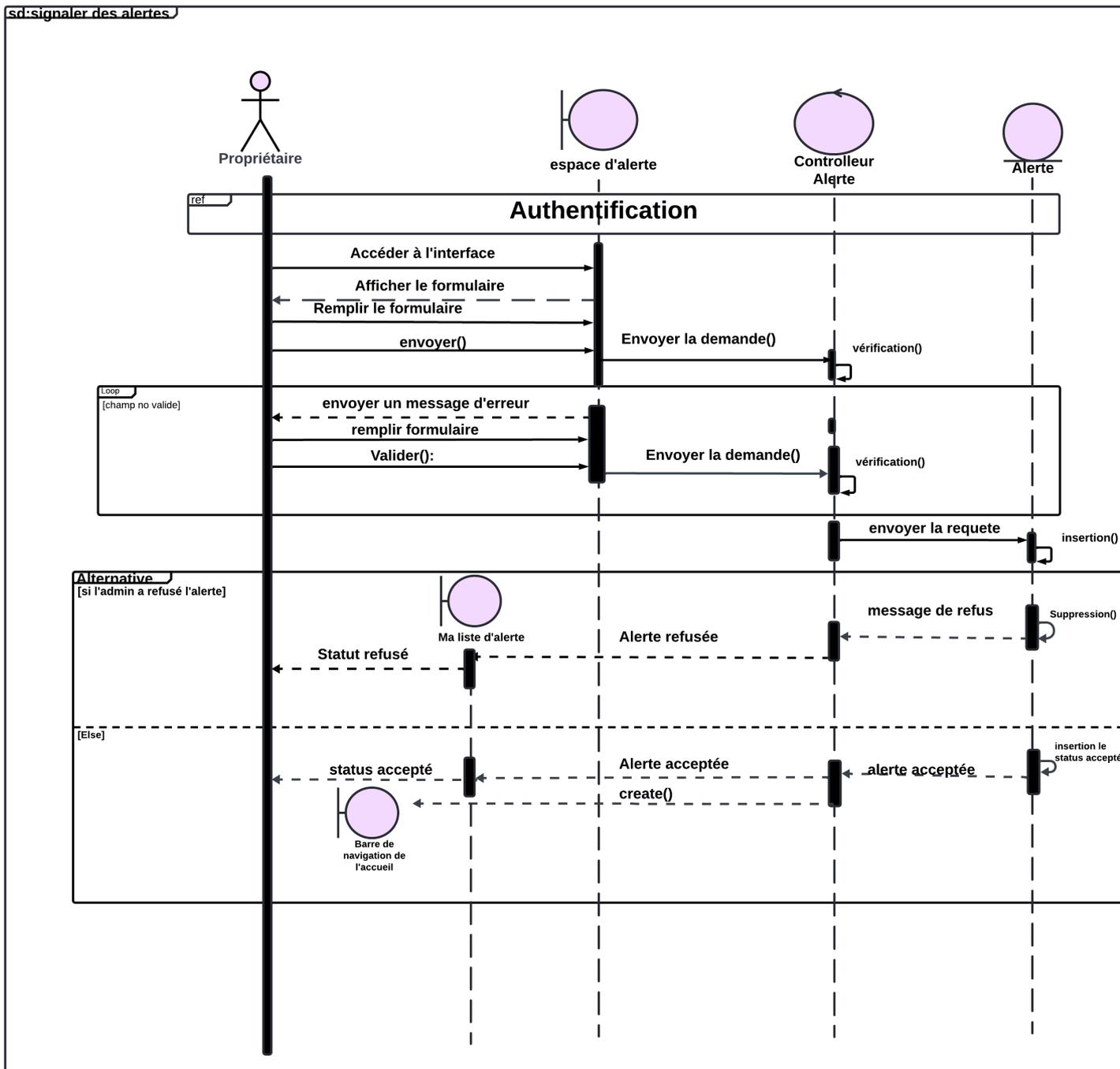


fig 4.2 : Diagramme d'interaction de cas d'utilisation 'Signaler des alertes'

4.2.3. Cas d'utilisation « Gestion des alertes »

la description textuelle du Cas d'utilisation « Gestion des alertes »

CU	Gestion des alertes
Acteur Principal	Administrateur
Objectif	Assurer la gestion efficace des alertes signalées par le propriétaire de l'animal
Précondition	L'administrateur doit être authentifié

<p>Scénario Principal</p>	<ol style="list-style-type: none"> 1. L'administrateur accède à l'espace gestion des alertes. 2. L'administrateur passe en revue les alertes soumises, en évaluant le contenu. 3. Pour chaque alerte, l'administrateur décide de l'accepter ou de la refuser. 4. Les alertes acceptées sont ensuite affichées sur la page d'accueil.
<p>Postcondition</p>	<p>Les alertes acceptées sont publiées au niveau de la barre de navigation de la page d'accueil et sont donc visibles par tous les utilisateurs.</p>

TABLE 4.2 – la description de Cas d'utilisation « Gestion des alertes »

Diagramme d'interaction de cas d'utilisation 'Gestion des alertes'

La figure suivante représente le diagramme d'interaction de cas d'utilisation 'Gestion des alertes'

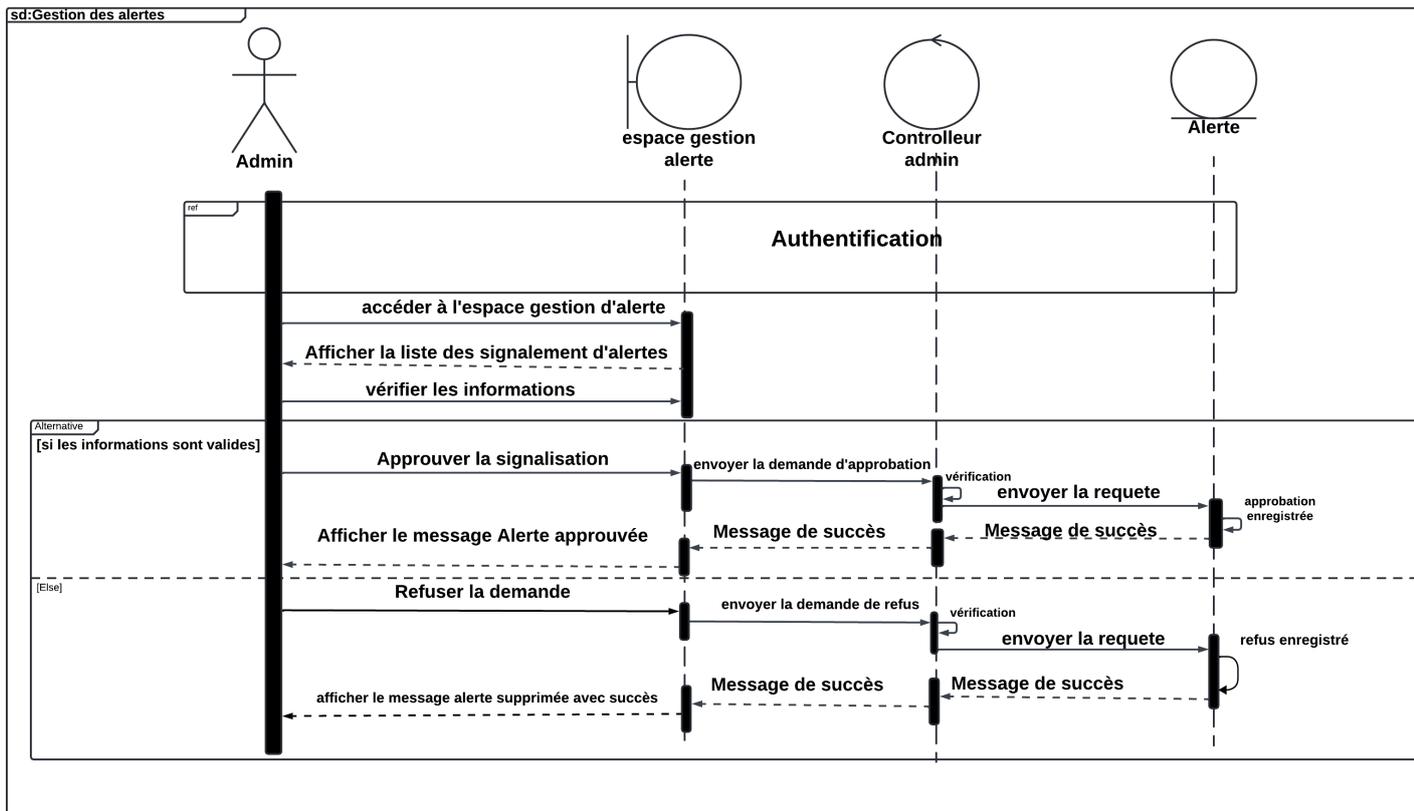


fig 4.3 : Diagramme d'interaction de cas d'utilisation 'Gestion des alertes'

4.2.4. Diagramme de classe du sprint 5

La figure suivante représente le diagramme de classe pour le sprint 5

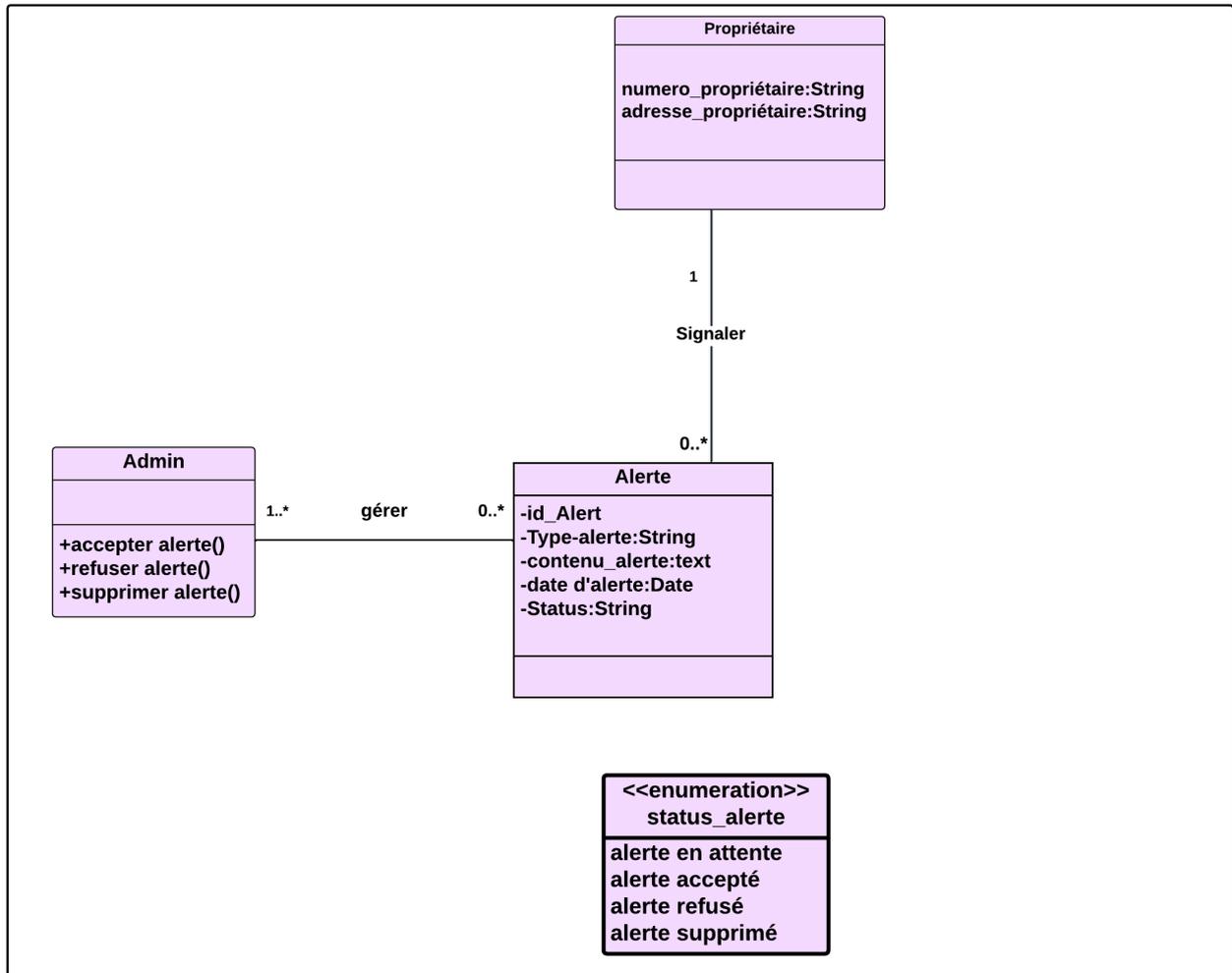


fig 4.4 : diagramme de classe sprint 05

4.3. Sprint 6 : la prise des rendez-vous

4.3.1. Introduction

Le sixième sprint de notre application introduit la fonctionnalité de prise de rendez-vous, permettant aux propriétaires d'animaux de trouver des vétérinaires par localisation. Les propriétaires peuvent visualiser des informations sur les vétérinaires et sélectionner celui qui correspond le mieux à leurs besoins. Après avoir rempli un formulaire de demande, le rendez-vous est soumis aux vétérinaires choisis. Le vétérinaire gère ensuite la demande, confirmant ou refusant le rendez-vous avec des explications si nécessaires.

4.3.2. Diagramme de cas d'utilisation du sprint 6

Le diagramme de cas d'utilisation ci-dessous illustre les cas d'utilisation correspondant aux items du sixième sprint :

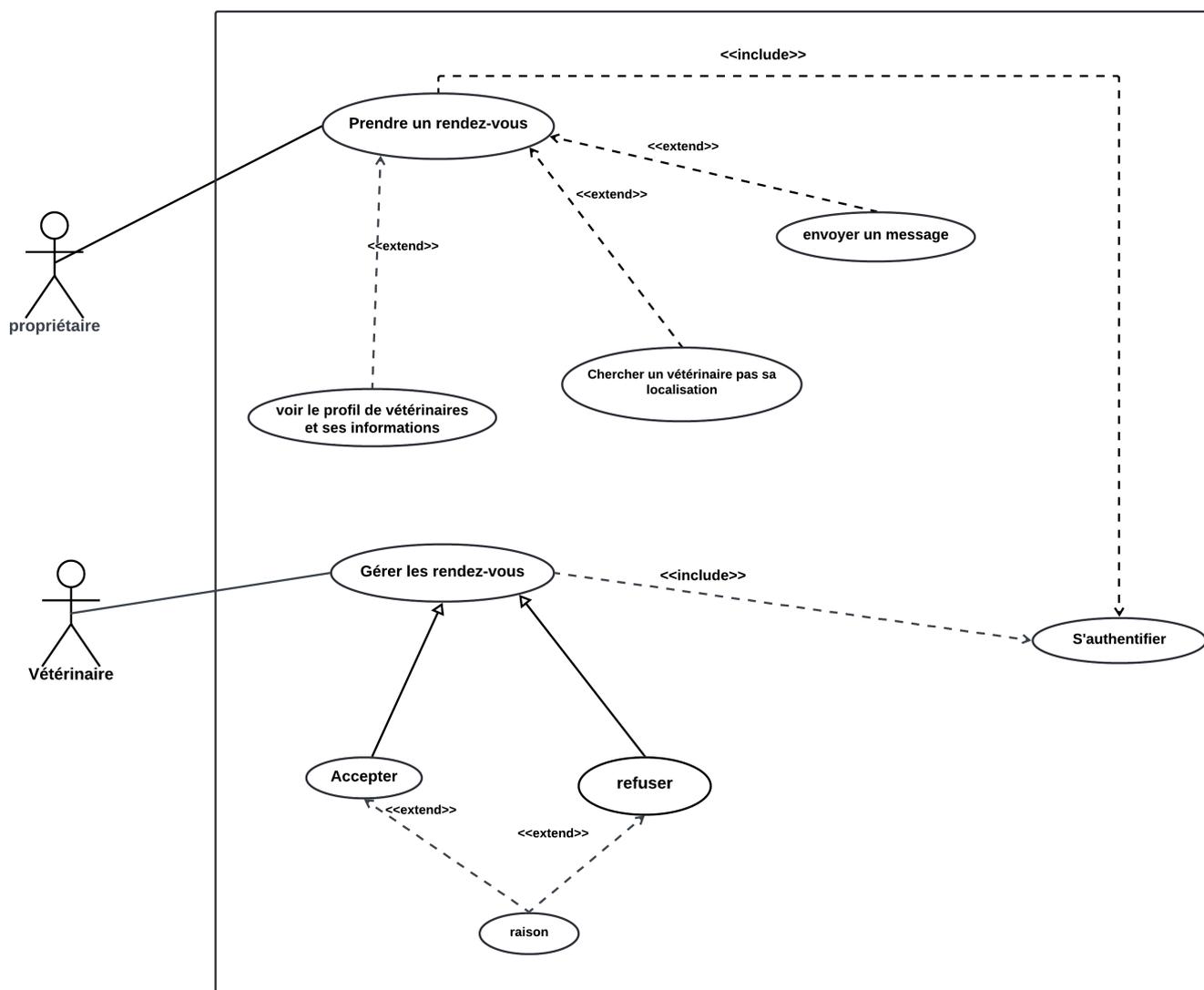


fig 4.5 : Diagramme de cas d'utilisation du sprint6.

4.3.3. Cas d'utilisation « consulter le profil et voir les informations du vétérinaire »

La description de Cas d'utilisation « consulter le profil et voir les informations du vétérinaire »

CU	consulter le profil et voir les informations du vétérinaire
Acteur Principal	Propriétaire de l'animal
Objectif	Permettre aux propriétaires d'accéder aux informations détaillées des vétérinaires et de les contacter si nécessaire
Précondition	<ul style="list-style-type: none"> — Le propriétaire est inscrit et connecté à son compte sur l'application. — Les profils des vétérinaires doivent être disponibles et à jour dans l'application.

Scénario Principal	<ol style="list-style-type: none">1. Le propriétaire accède à la liste des vétérinaires dans l'application.2. Il peut utiliser la fonction de recherche pour filtrer un vétérinaire selon la localisation.3. Il sélectionne un vétérinaire pour consulter son profil.4. Le profil du vétérinaire s'affiche, incluant les informations telles que le nom, la localisation, les heures de travail, etc.5. Si le propriétaire le souhaite, il peut initier le contact avec le vétérinaire via l'espace de messagerie.
Postcondition	Le propriétaire accède aux informations du vétérinaire et a la possibilité de le contacter.
Cas d'utilisation alternatifs	Si le vétérinaire n'est pas disponible, le propriétaire reçoit un message indiquant que le vétérinaire n'est pas disponible.

TABLE 4.3 – la description de cas d'utilisation « consulter le profil et voir les informations du vétérinaire »

Diagramme d'interaction de cas d'utilisation « rechercher un vétérinaire » :

La figure suivante présente le diagramme d'interaction de cas d'utilisation 'rechercher un vétérinaire '

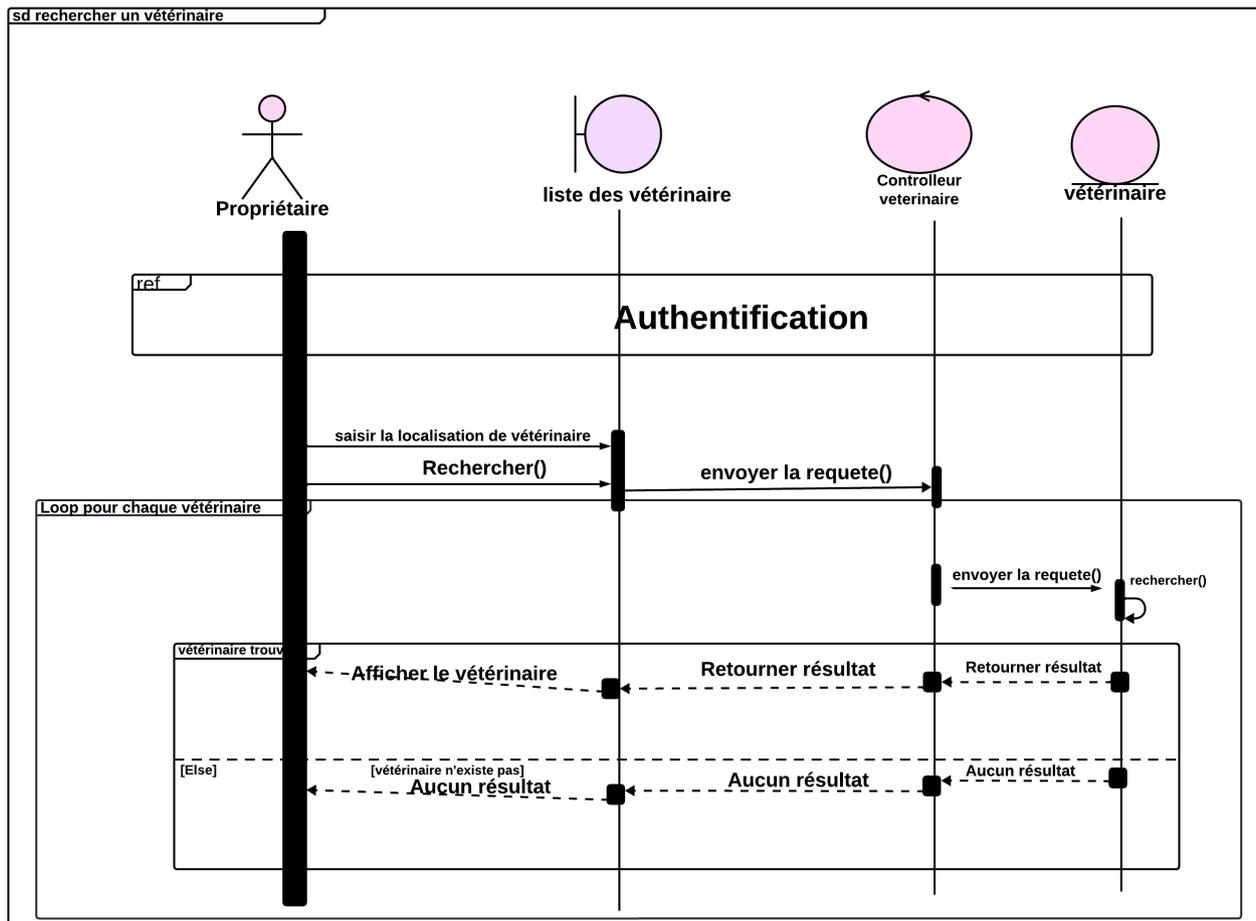


fig 4.6 : Diagramme d'interaction de cas d'utilisation « rechercher un vétérinaire ».

4.3.4. Cas d'utilisation « Prendre un rendez-vous »

La description textuelle du Cas d'utilisation « Prendre un rendez-vous »

CU	Prendre un rendez-vous
Acteur Principal	Propriétaire
Objectif	Faciliter la prise de rendez-vous entre le propriétaire et le vétérinaire

Scénario Principal	<ol style="list-style-type: none"> 1. Le propriétaire accède à l'espace des vétérinaire. 2. Le propriétaire peut rechercher un vétérinaire en fonction de sa localisation. 3. Il consulte les informations et les disponibilités des vétérinaires. 4. Il sélectionne le vétérinaire avec lequel il souhaite prendre un rendez-vous. 5. La demande de rendez-vous est envoyée au vétérinaire sélectionné. 6. Le système affiche un message confirmant que la demande a été envoyée avec succès.
Postcondition	Le rendez-vous est planifié et confirmé entre le propriétaire et le vétérinaire.
Cas d'utilisation alternatifs	<ul style="list-style-type: none"> — Si le vétérinaire refuse le rendez-vous, il fournit une raison pour le refus et propose un autre créneau. — Le propriétaire peut choisir d'accepter le nouveau créneau ou de chercher un autre vétérinaire.

TABLE 4.4 – description de cas d'utilisation « prendre un rendez vous »

Diagramme d'interaction de cas d'utilisation « prendre un rendez vous » :

La figure suivante représente le diagramme d'interaction du cas d'utilisation 'prendre un rendez vous'

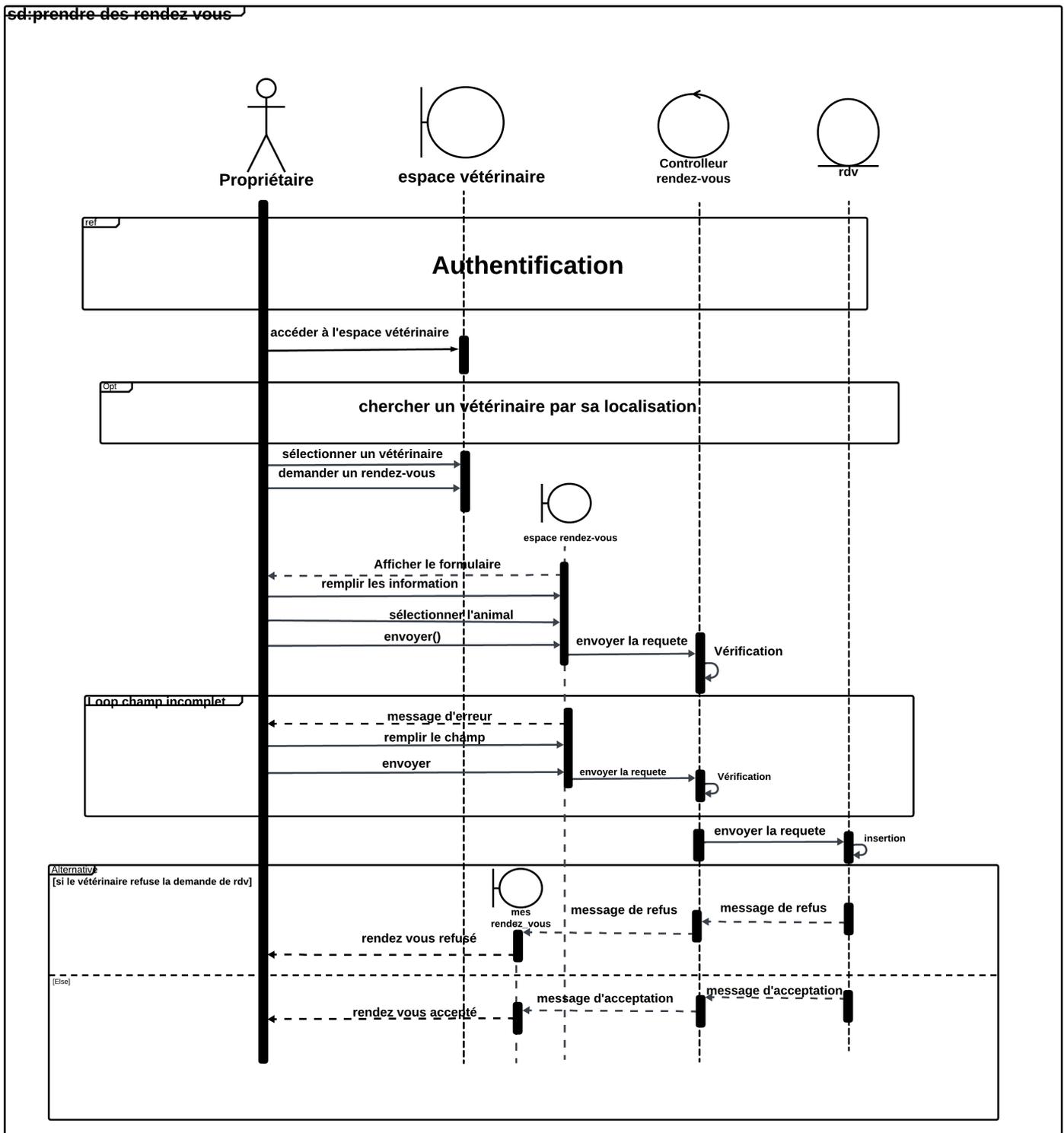


fig 4.7 : Diagramme d'interaction de cas d'utilisation prendre rendez vous .

4.3.5. Cas d'utilisation « Gestion des rendez-vous »

La description textuelle du Cas d'utilisation « Gestion des rendez-vous »

CU	Gestion des rendez-vous
Acteur Principal	Vétérinaire
Objectif	Informé le propriétaire de la disponibilité du vétérinaire

Précondition	Le vétérinaire doit être authentifié
Scénario Principal	<ol style="list-style-type: none">1. Le vétérinaire accède à l'espace de gestion de rendez-vous.2. Le vétérinaire vérifie les rendez-vous (date, heure, etc.).3. Le vétérinaire accepte ou refuse le rendez-vous.4. Si un rendez-vous est supprimé, le vétérinaire fournit une raison de refus.
Postcondition	La décision du vétérinaire sera affichée dans l'espace propriétaire.

TABLE 4.5 – description de cas d'utilisation gestion rendez-vous

Diagramme d'interaction de cas d'utilisation « gestion des demande de rendez vous » :

La figure suivante présente le diagramme d'interaction de cas d'utilisation 'gérer les demandes de rendez-vous '

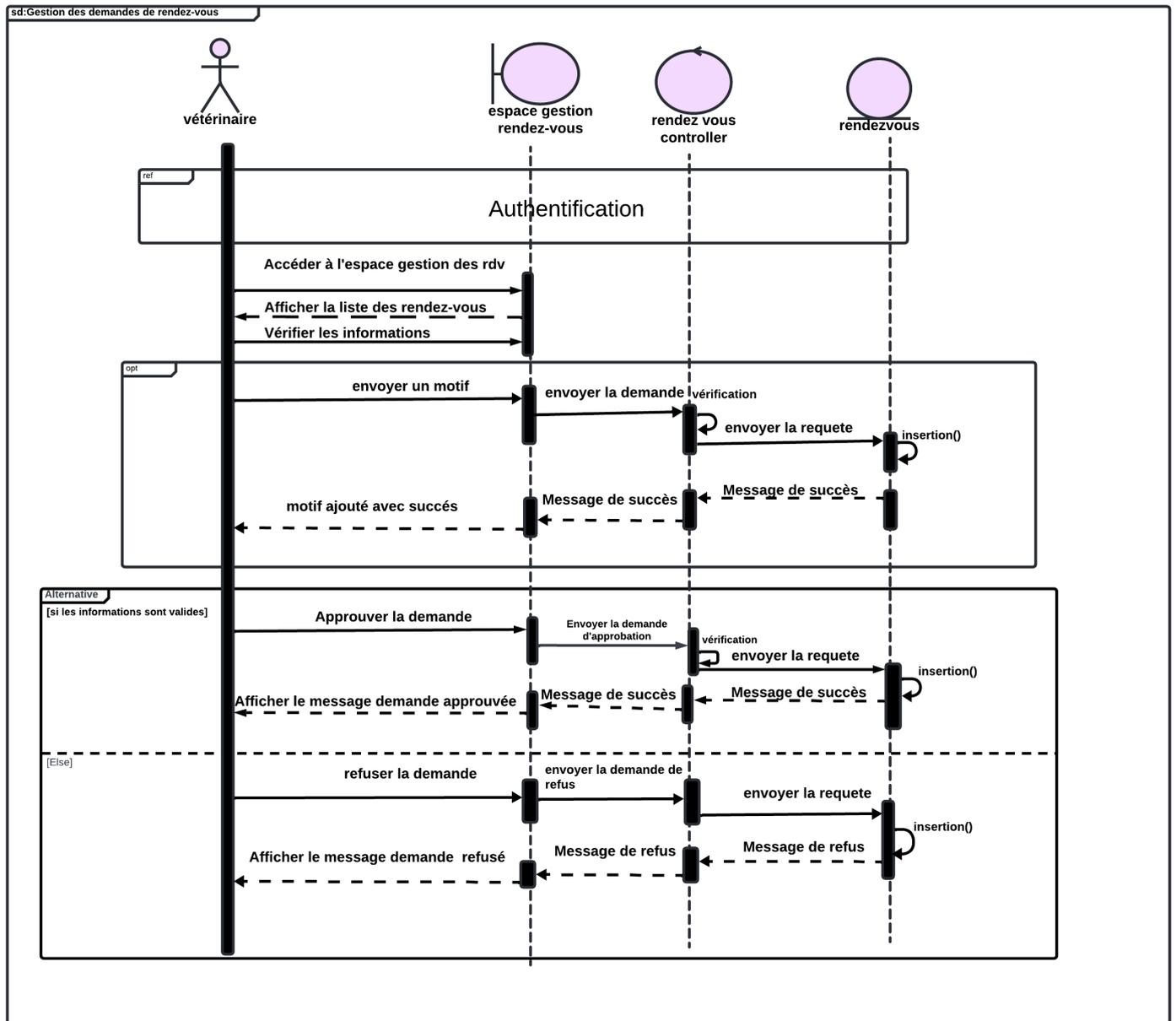


fig 4.8 : Diagramme d'interaction de cas d'utilisation gérer les demandes de rendez-vous .

4.3.6. Diagramme de classe du sprint6

La figure suivante représente le diagramme de classe pour le sprint 6

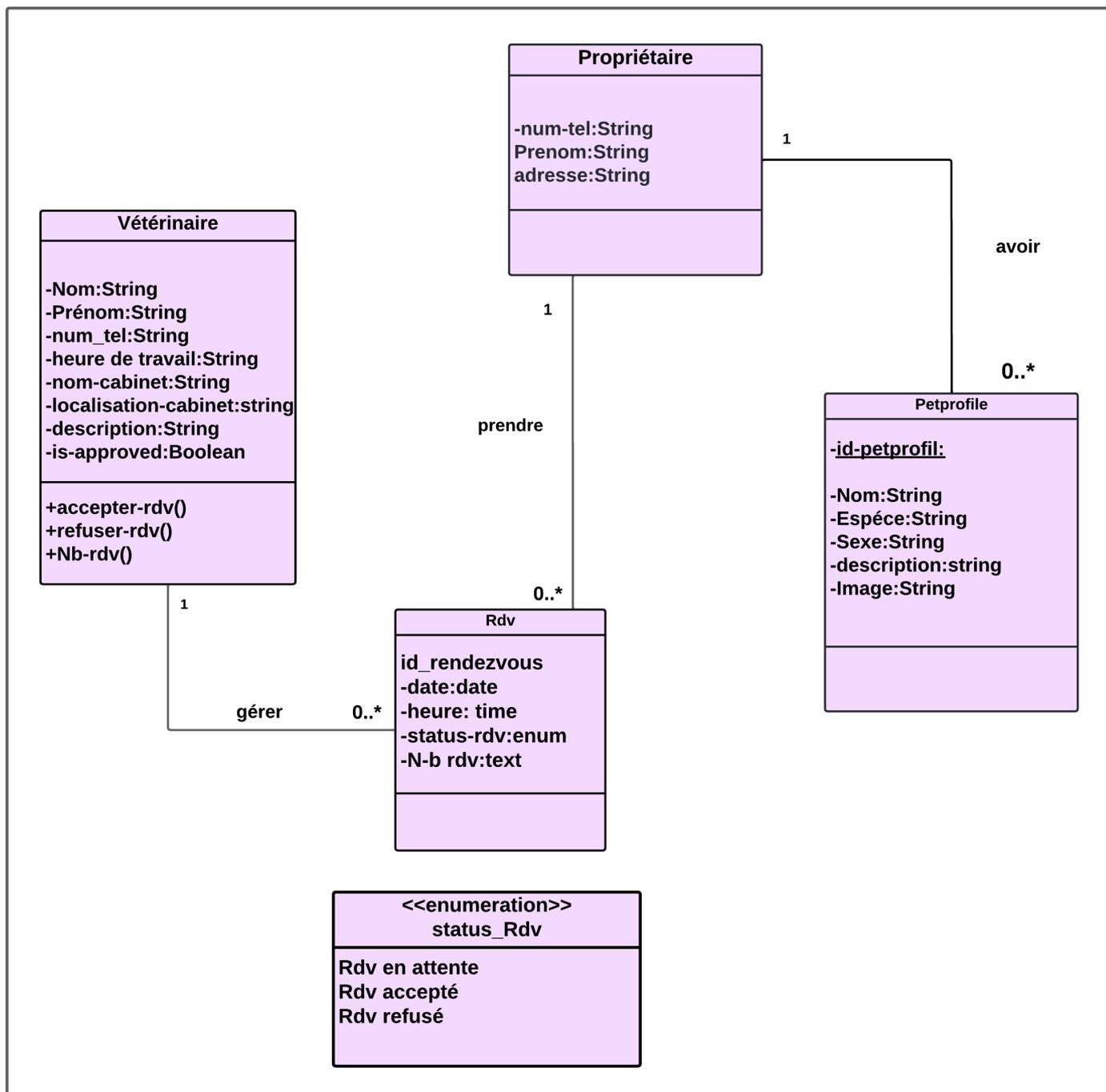


fig 4.9 : Diagramme de classe du sprint6 .

4.4. Conclusion

Notre deuxième release représente une avancée significative dans le développement de notre projet, marqué par la réussite de deux sprints essentiels. Le sprint de « Signalement des alertes » a introduit une fonctionnalité permettant aux utilisateurs de signaler diverses urgences liées aux animaux, telles que les pertes des animaux ou les besoins urgents. Parallèlement, le sprint de « Prise de rendez-vous » a simplifié la coordination entre les propriétaires et les vétérinaires. De plus, les lacunes identifiées au premier release ont été comblé avec succès, consolidant ainsi notre progression vers une application plus solide et fonctionnelle. Prochainement nous allons entamer la phase du dernier release.

Chapitre 5

Troisième Release

5.1. Introduction

Dans ce chapitre, nous allons présenter le travail effectué durant le troisième release qui comprend :

- **Le sprint 7 : L'adoption**
- **Le sprint 8 : Publication**

Pour chacun de ces sprints, nous avons élaboré des diagrammes de cas d'utilisation ainsi que des descriptions textuelles pour les cas d'utilisation et des diagrammes d'interactions.

5.2. sprint 7 : « L'adoption »

Le septième sprint implémente un système permettant aux propriétaires de proposer des animaux à adopter dans l'espace adoption. Les autres propriétaires peuvent alors adopter un ou plusieurs animaux, et le propriétaire initial gère les demandes d'adoption. Ce sprint est réalisé dans une semaine, il comprend deux items :

- Afficher des animaux à adopter.
- Demande d'adoption d'animaux.

5.2.1. Diagramme de cas d'utilisation du sprint 7

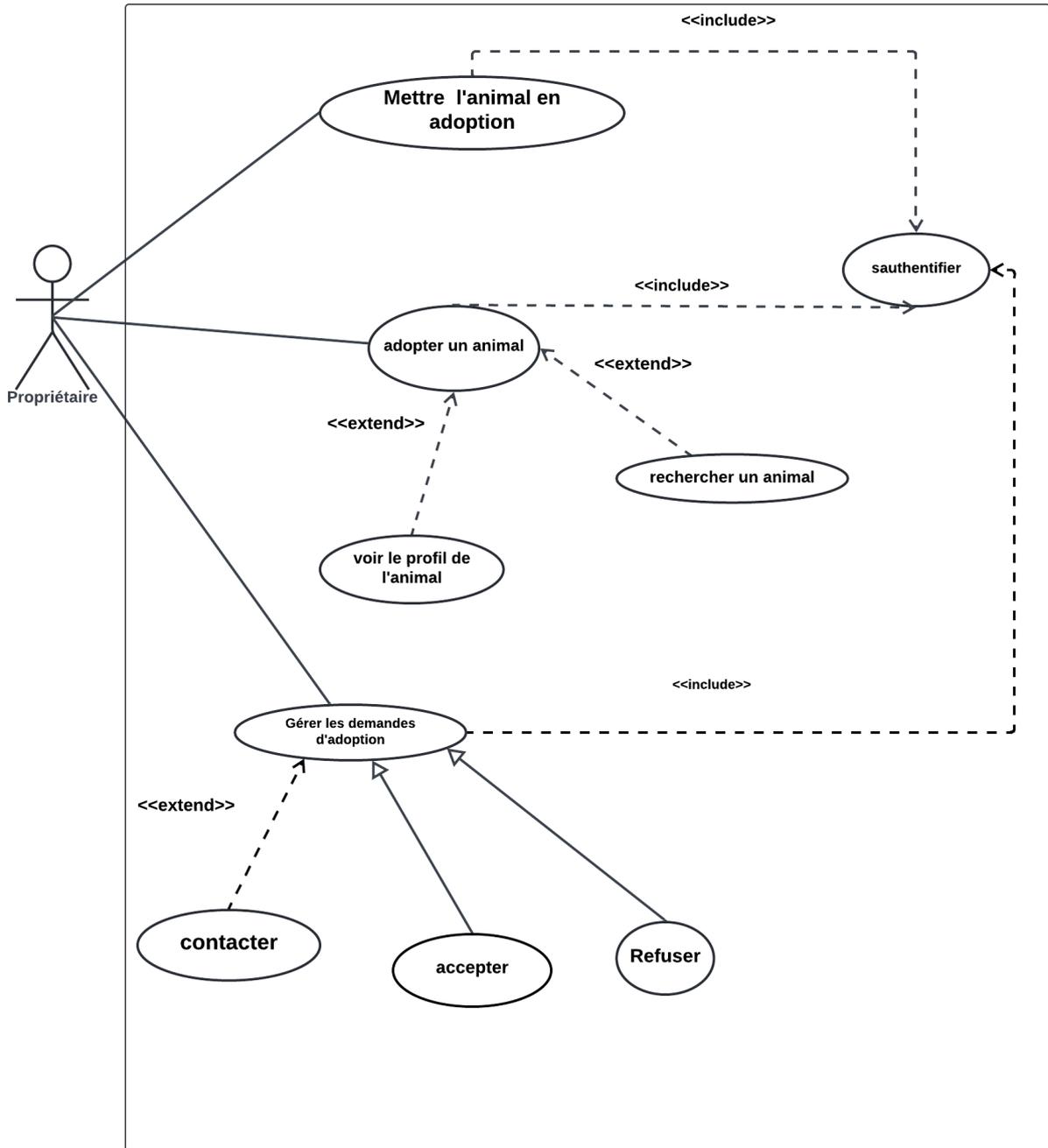


fig 5.1 : Diagramme de cas d'utilisation du sprint 7

5.2.2. Cas d'utilisation « Adopter un animal »

Titre	Adopter un animal
Acteur Principal	Adoptant, Propriétaire

Conditions	<ul style="list-style-type: none"> — l’adoptant ou le propriétaire doit être inscrit et connecté à l’application. — Des annonces d’animaux disponibles pour adoption doivent être publiées.
Scénario Principal	<ol style="list-style-type: none"> 1. L’utilisateur navigue vers la section « Adoption » de l’application. 2. L’utilisateur utilise un filtre de recherche pour trouver les animaux disponibles pour l’adoption. 3. L’utilisateur sélectionne l’animal qui l’intéresse. 4. L’utilisateur consulte le profil de l’animal pour voir ses informations détaillées. 5. Si l’utilisateur est intéressé, il clique sur le bouton « Adopter Animal ». 6. L’utilisateur soumet la demande d’adoption. 7. L’application envoie la liste de demandes au propriétaire actuel de l’animal pour confirmer ou supprimer la demande. 8. Si le propriétaire accepte, l’adoption est finalisée et les deux parties organisent la remise de l’animal dans l’espace messagerie.
Extensions	Si l’utilisateur a des questions, il peut contacter le propriétaire.

TABLE 5.1 – description de cas d’utilisation adopter animal"

Diagramme d'interaction du cas d'utilisation « Adopter un animal »

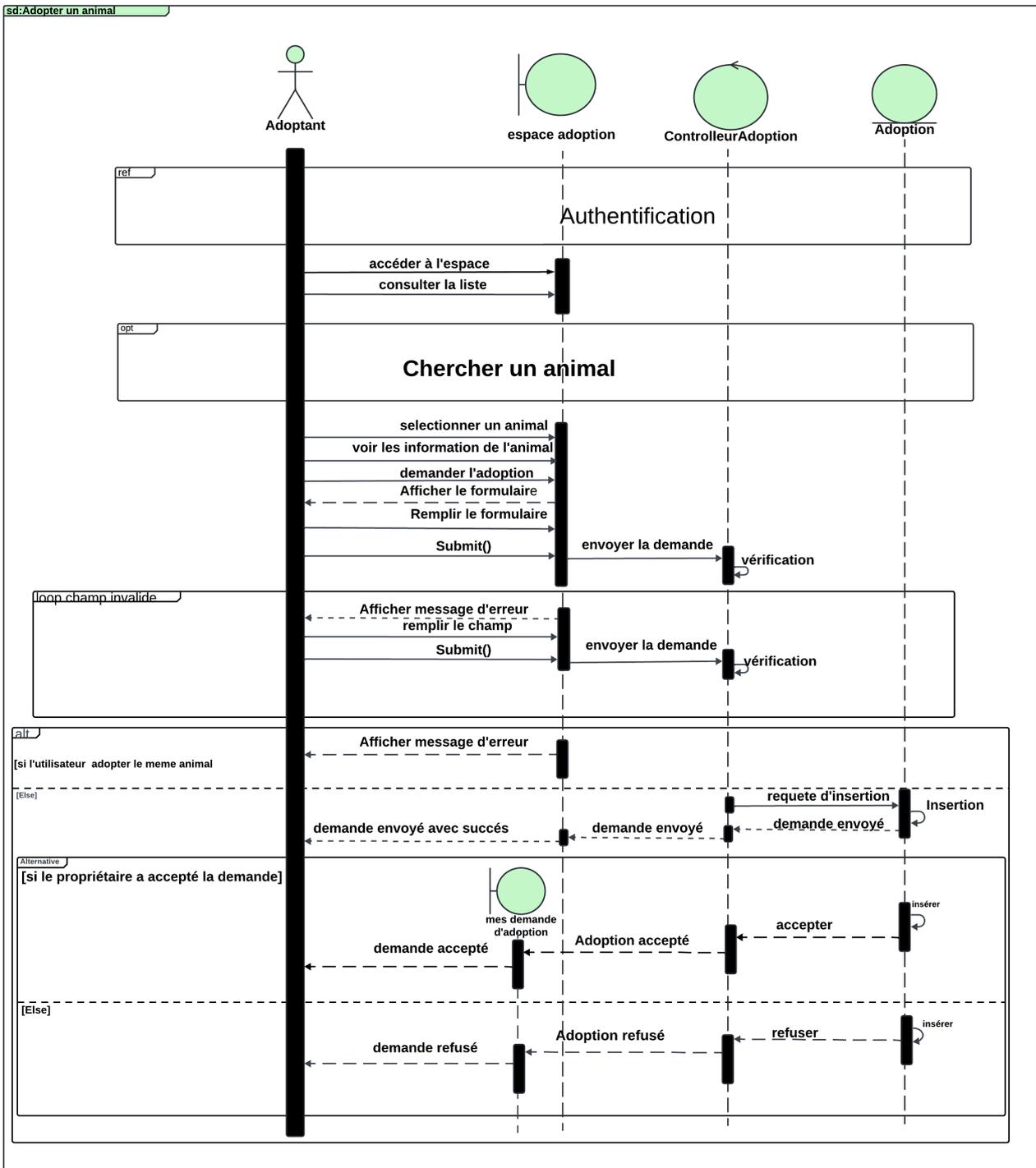


fig 5.2 : Diagramme d'interaction du cas d'utilisation « Adopter un animal »

5.2.3. Cas d'utilisation « Gestion des adoptions »

Titre	Gestion des adoptions
Acteur Principal	Propriétaire de l'animal

<p>Préconditions</p>	<ul style="list-style-type: none"> — Le propriétaire doit s’authentifier. — Le propriétaire doit avoir listé l’animal comme disponible pour l’adoption.
<p>Scénario Principal</p>	<ol style="list-style-type: none"> 1. Le propriétaire accède à la section « Gérer les Adoptions ». 2. Le propriétaire examine les demandes d’adoption reçues pour son animal. 3. Le propriétaire sélectionne une demande pour l’approuver ou la supprimer. 4. Le propriétaire engage la conversation avec le futur propriétaire pour discuter les détails et les conditions d’adoption. 5. Le statut de l’animal est mis à jour comme « adopté ».

TABLE 5.2 – description de cas d’utilisation « Gestion des adoptions »

Diagramme d’interaction du cas d’utilisation « Gestion des adoptions »

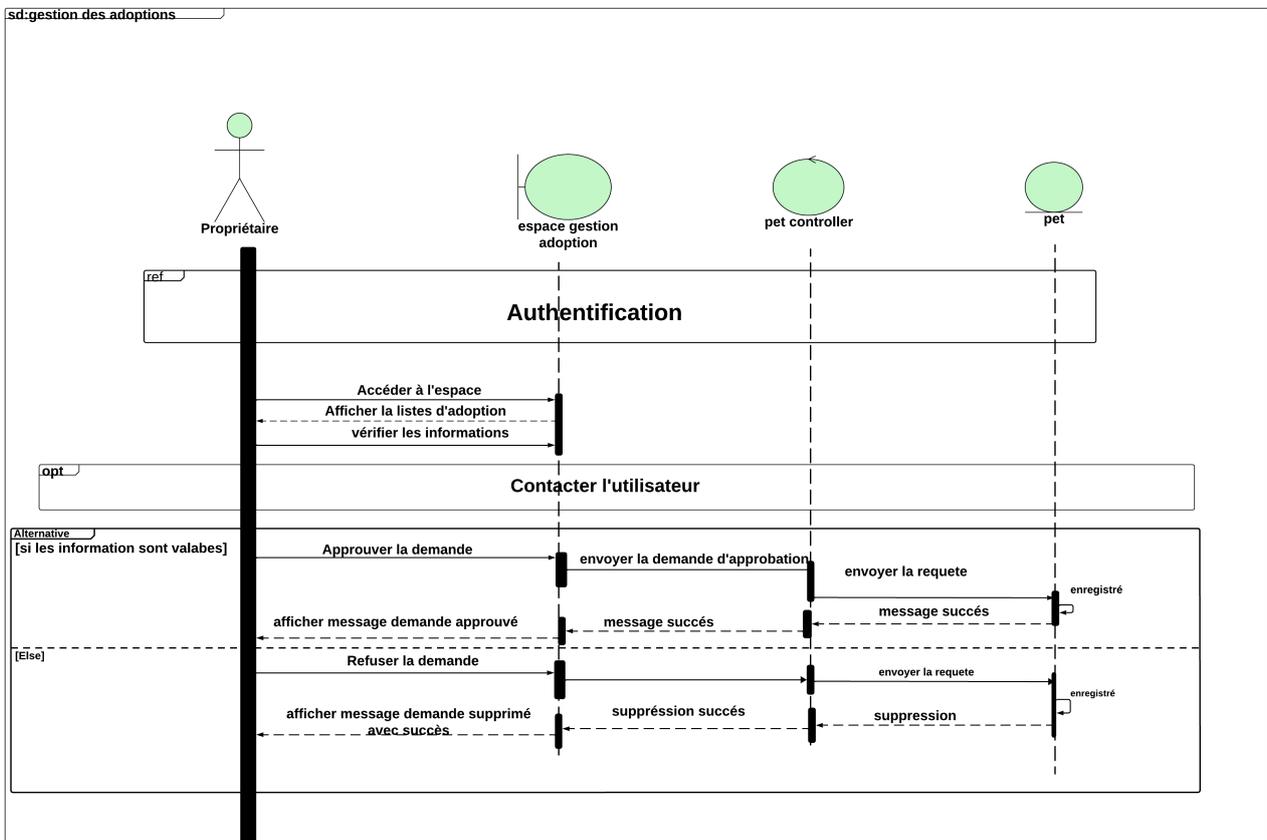


fig 5.3 : Diagramme d’interaction du cas d’utilisation « Gestion des adoptions »

5.2.4. Diagramme de classe de sprint 7

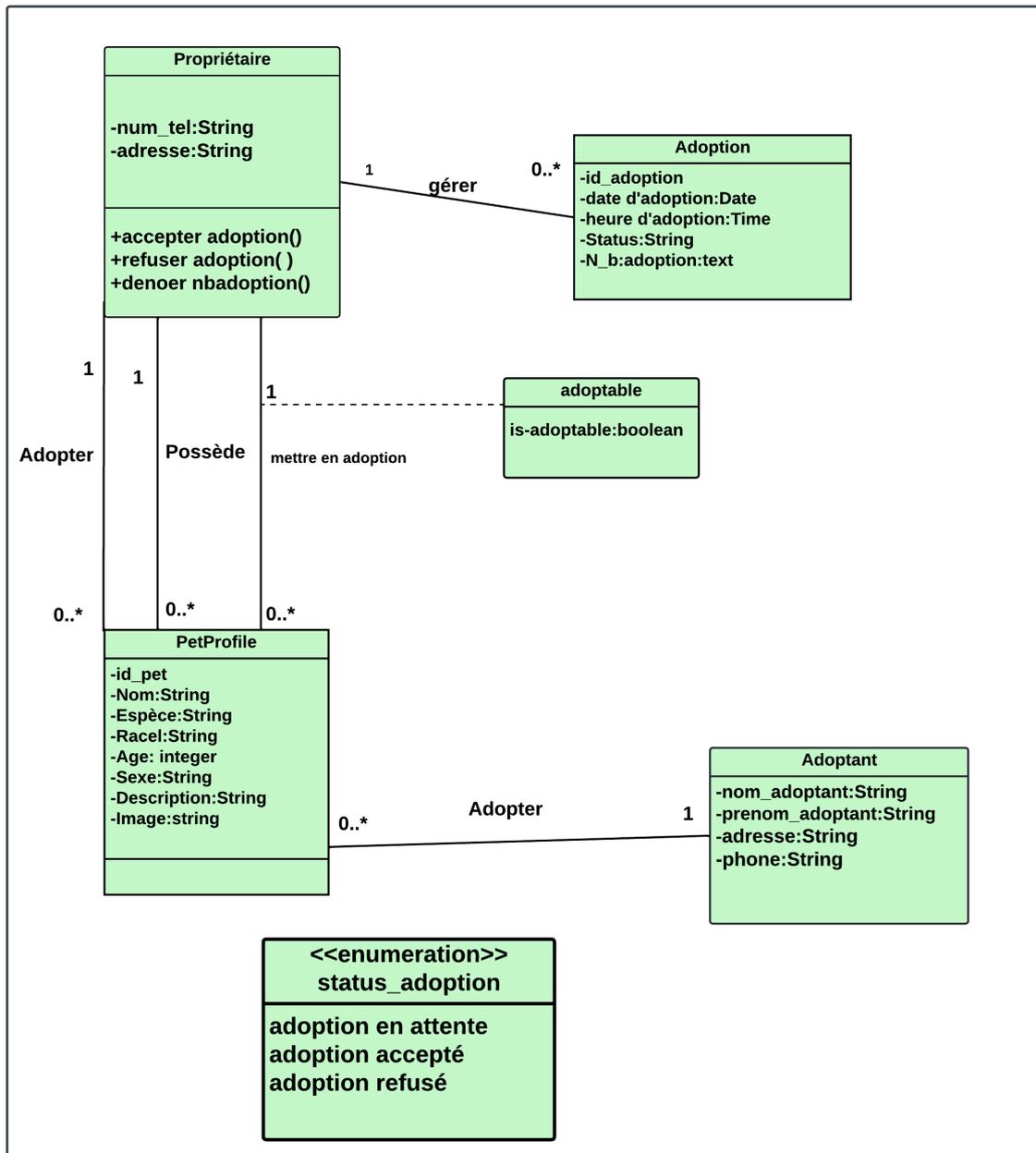


fig 5.4 : Diagramme de classe « sprint7 »

5.3. Sprint 8 : « Publication »

Le sprint de publication est une itération de travail durant laquelle les fonctionnalités permettent aux utilisateurs de publier du contenu, interagir avec celui-ci via des « j’aime » et des commentaires. En outre, il inclut la possibilité de signaler les publications inappropriées. Ce sprint est réalisé dans deux semaines et comprend trois items :

- Effectuer des publications.
- Interactions avec les publications (j’aime, commentaire, etc.).
- Le signalement des publications.

5.3.1. Diagramme de cas d'utilisation du sprint 8 « Publication »

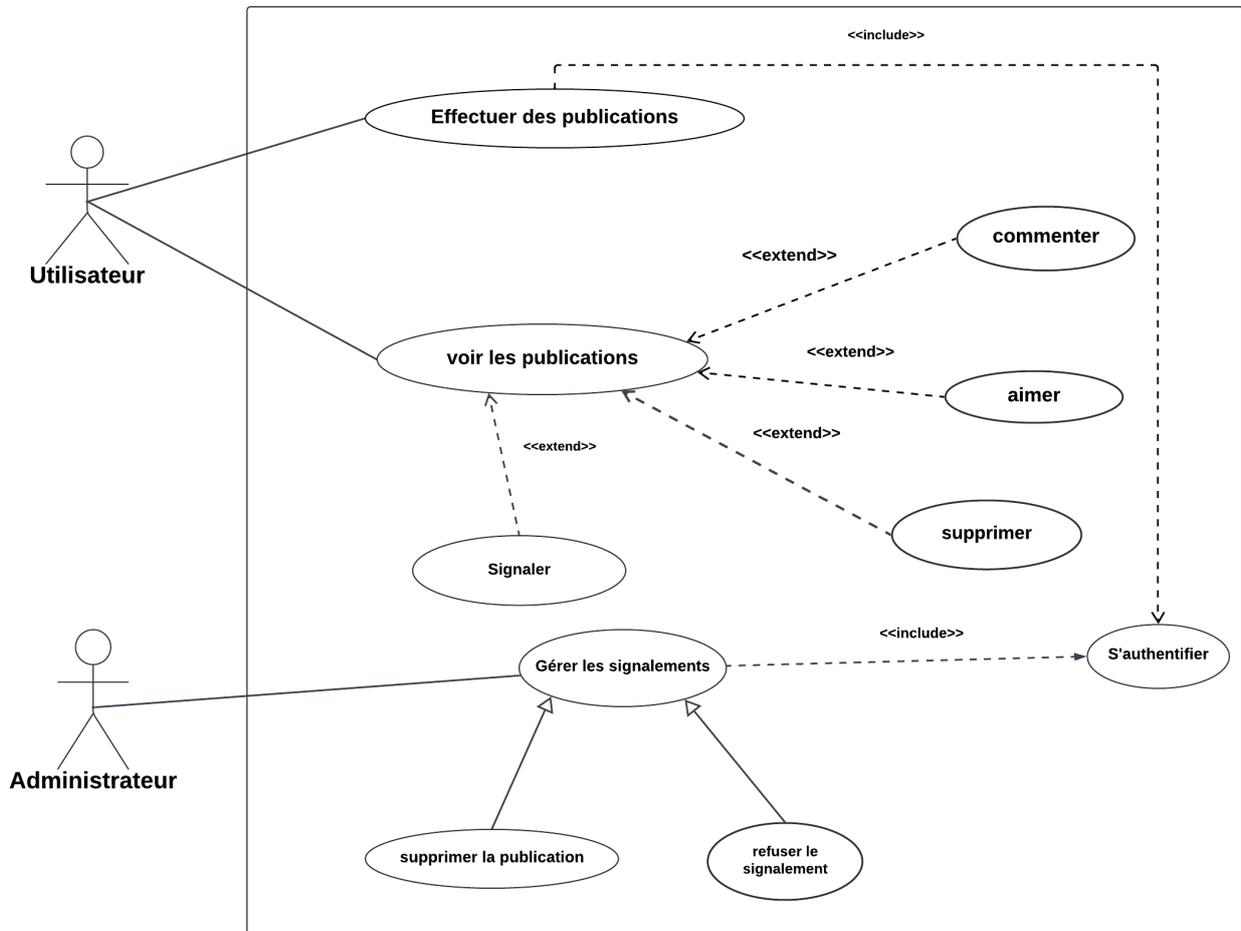


fig 5.5 : Diagramme de cas d'utilisation du sprint 8

5.3.2. Cas d'utilisation « Effectuer des publications »

Titre	Effectuer des publications
Acteur Principal	Utilisateur de l'application
Précondition	— L'utilisateur doit s'authentifier.
Scénario Principal	<ol style="list-style-type: none"> 1. L'utilisateur accède à l'espace publication. 2. L'utilisateur rédige la publication. 3. L'utilisateur publie le contenu sur la plateforme. 4. Les autres utilisateurs pourront voir et interagir avec la publication via des « j'aime » et des commentaires.

Postcondition	<ul style="list-style-type: none"> — Nouvelle publication créée. — Les autres utilisateurs peuvent interagir avec la publication.
Extension	En cas de contenu inapproprié, les utilisateurs pourront signaler la publication.

TABLE 5.3 – description de Cas d'utilisation « Effectuer des publications »

Diagramme d'interaction du cas d'utilisation « effectuer des publications »

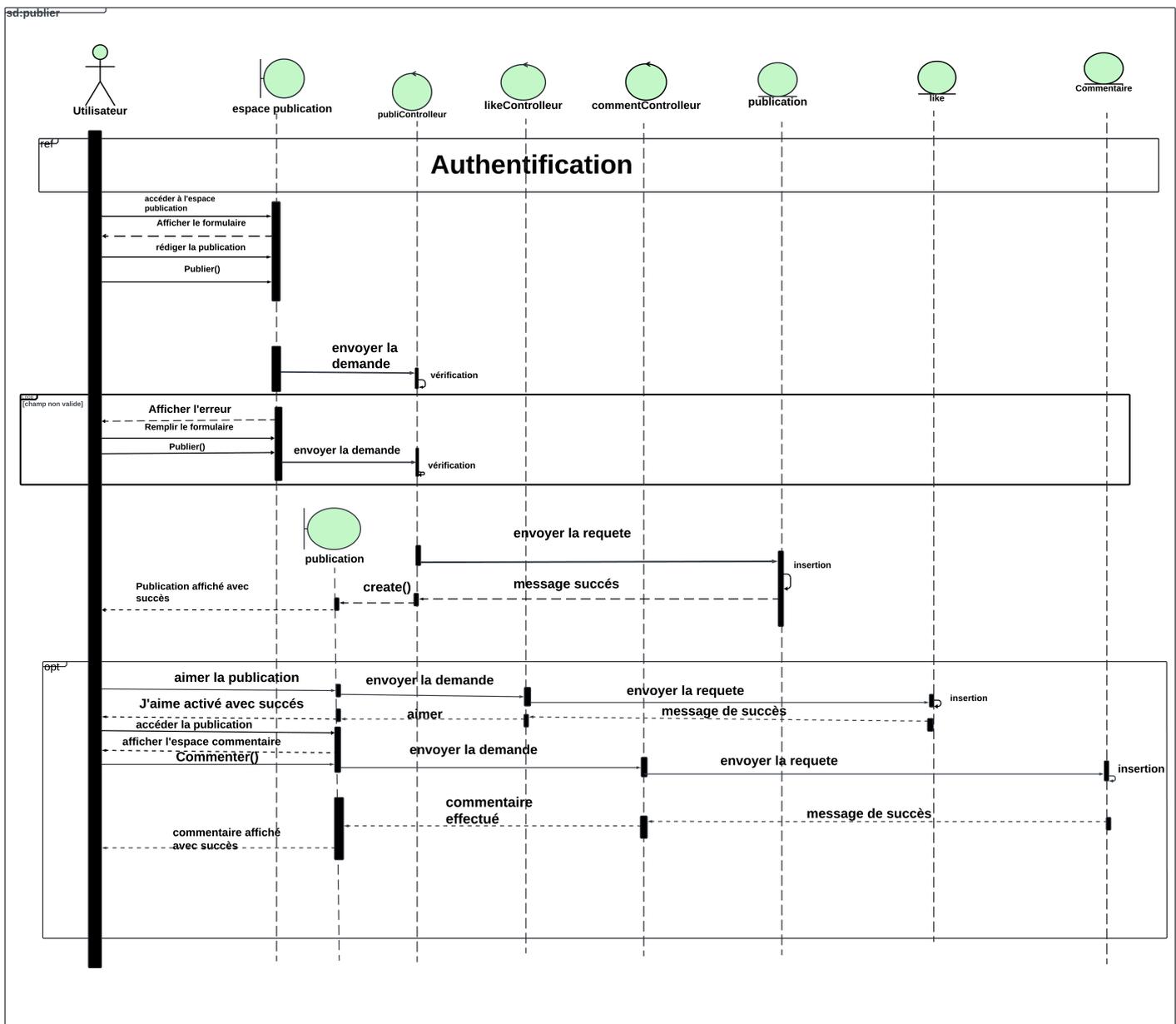


fig 5.6 : Diagramme d'interaction du cas d'utilisation « Effectuer des publications »

5.3.3. Cas d'utilisation « signaler des publications »

La description textuelle du Cas d'utilisation « signaler des publications »

Titre	sihner des publications
Acteur Principal	l'utilisateur(propriétaire,vétérinaire,adoptant)
Précondition	— L'utilisateur doit s'authentifier.
Scénario Principal	<ol style="list-style-type: none"> 1. L'utilisateur accède à l'espace publication. 2. L'utilisateur signale la publication. 3. Le système lui affiche un formulaire . 4. L'utilisateur doit remplir les information de signalement ainsi la cause 5. La demande ca sera envoyé aux admins de l'application pour accepter ou refuser le signalement
Postcondition	— la publication ça sera supprimé si l'admin a accepté le signalement .
Extension	En cas de contenu inapproprié, les utilisateurs pourront signaler la publication.

TABLE 5.4 – description de Cas d'utilisation « signaler des publications »

Diagramme d'interaction du cas d'utilisation « signaler des publications »

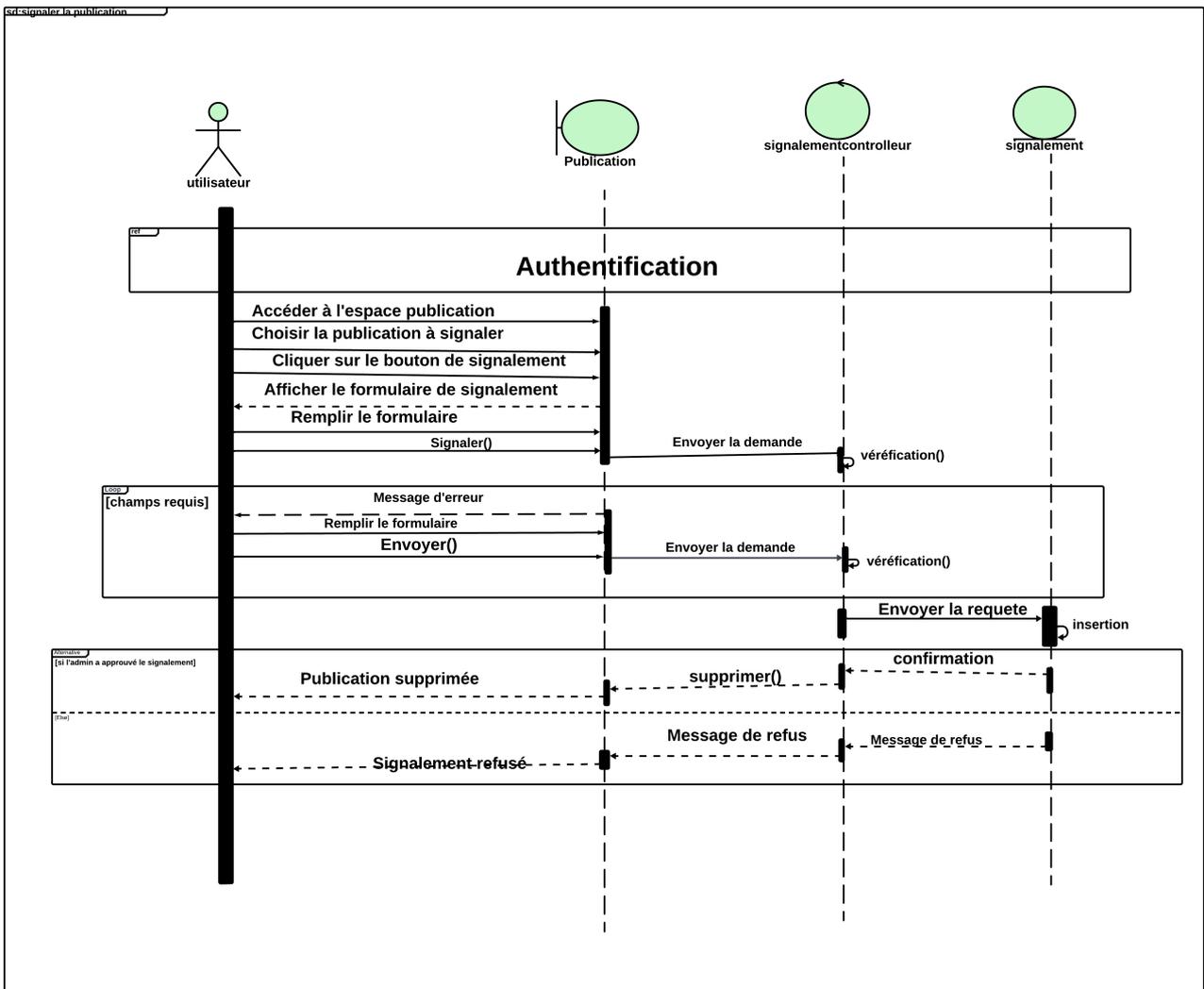


fig 5.7 : Diagramme d'interaction du cas d'utilisation «signaler la publication»

5.3.4. Diagramme de classe de sprint 8

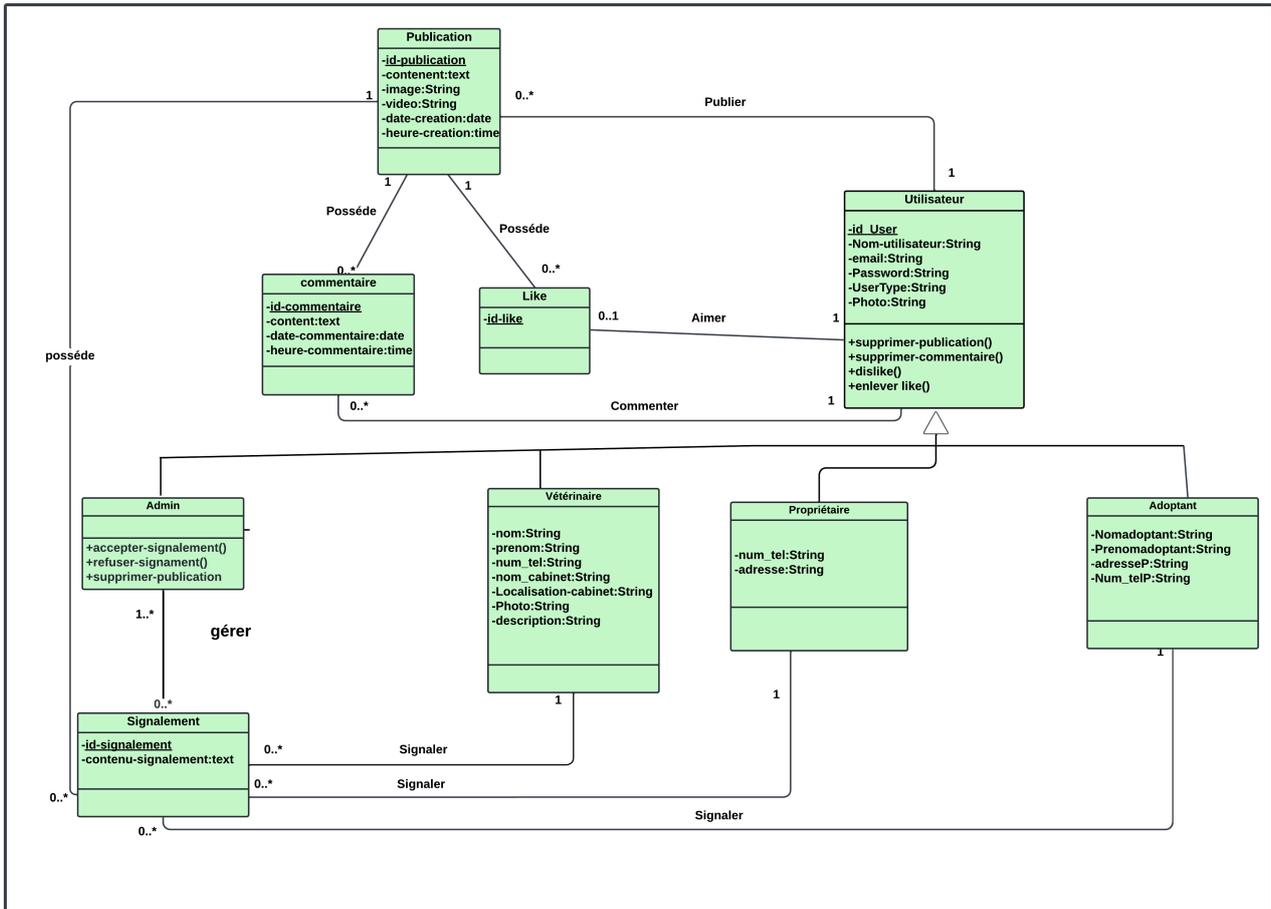


fig 5.8 : Diagramme de classe «sprint8»

5.4. Conclusion

Le troisième release a étendu les fonctionnalités de l'application en intégrant la gestion des adoptions d'animaux et la gestion des publications. Le septième Sprint a ajouté la capacité pour les utilisateurs de mettre en adoption leurs propres animaux, et pour les utilisateurs souhaitant en adopter, de soumettre des demandes d'adoption. Le huitième Sprint a introduit la possibilité pour les utilisateurs de créer et partager du contenu sur la plateforme, avec des interactions sous forme de « j'aime », de commentaires, et de signalements pour les contenus inappropriés. Ces fonctionnalités ont enrichi l'interaction sociale sur la plateforme et renforcé la gestion des contenus et des adoptions, marquant ainsi une étape clé dans le développement de l'application. L'étape suivante va aborder la phase de la modélisation de base de données.

Chapitre 6

Modélisation de la Base de données

6.1. Introduction

Ce chapitre aborde la conception de la base de données, un élément crucial pour le bon fonctionnement de notre application. Après avoir défini les besoins et les fonctionnalités à travers les sprints Scrum précédents, nous nous concentrons sur l'établissement des règles de gestion. Ces règles sont fondamentales pour structurer et organiser les données, garantissant ainsi l'intégrité et la cohérence du système, et guideront la création du diagramme de classe, du dictionnaire de données et du modèle relationnel, essentiels pour la mise en œuvre technique.

6.2. Règles de gestion

Afin d'élaborer un diagramme de classe compréhensible et adapté aux besoins du système Petrack, nous avons suivi les règles de gestion suivantes :

- Un utilisateur peut être soit un propriétaire, un vétérinaire, un adoptant ou un administrateur.
- Un propriétaire peut créer plusieurs profils pour ses animaux domestiques.
- Un propriétaire peut gérer les comptes de ses animaux (modifier, supprimer).
- Un animal peut avoir une ou plusieurs notes.
- chaque propriétaire peut signaler plusieurs alertes.
- L'administrateur peut gérer plusieurs alertes.
- Le vétérinaire peut ouvrir un seul compte.
- L'administrateur peut gérer plusieurs demandes d'ouverture de compte vétérinaire.
- chaque propriétaire peut prendre plusieurs rendez-vous.
- Le rendez-vous est pris par un seul vétérinaire.
- chaque vétérinaire gère ses rendez-vous.
- L'utilisateur peut contacter plusieurs utilisateurs.
- chaque propriétaire peut mettre en adoption plusieurs animaux.
- L'adoptant ou le propriétaire peut adopter un ou plusieurs animaux.
- L'utilisateur peut publier plusieurs publications.
- L'utilisateur peut aimer plusieurs publications, chacune est peut-être aimée qu'une seule fois.
- L'utilisateur peut écrire plusieurs commentaires dans une publication.
- L'administrateur ou vétérinaire ou adoptant peut signaler plusieurs publications.
- L'administrateur peut gérer plusieurs signalements.

6.3. Diagramme de classe

Un diagramme de classe est essentiel en modélisation orientée objet (UML), fournissant une représentation statique de la structure interne du système en définissant les types d'objets et leurs relations, indépendamment de tout langage de programmation. Il illustre la vue statique qui est les classes et leurs relations comportant [20] :

- Les associations : décrit les connexions entre les instances de deux classes ou plus, représentée par un trait avec le nom de la relation et la cardinalité de chaque classe.
- Les agrégations : la classe composante peut être retirée sans affecter la classe composée.
- Les compositions : Retirer la classe composée entraîne également la suppression de la classe composante.
- Les généralisations : Représente la relation d'héritage, où la classe héritée hérite des attributs et méthodes de la classe mère [20].

La figure suivante représente le diagramme de classe :

	usertype	Type d'utilisateur	String	(admin, user, vétérinaire)
	photo	Photo	String	
Propriétaire	adresse num_telephone	Adresse Numéro de téléphone	String String(10)	
Adoptant	nomAdopt prénomAdopt adresse num_telephone	Nom adoptant Prénom adoptant Adresse Numéro de téléphone	String(10) String(10) String String(10)	
Vétérinaire	nomveto prénomveto numtel heure_travail nom_cabinet localisation description isapproved image	Nom de vétérinaire Prénom de vétérinaire Numéro de téléphone Heure de travail nom de cabinet localisation de cabinet Description Approbation Image	String String String(10) String String(20) String(4) Text(255) Boolean String	(0,1)
PetProfil	id_pet nom espèce race âge sexe description isAdoptable image	Identifiant de l'animal Nom de l'animal Espèce Race Âge Sexe Description Animal adoptable String	BigInteger String(String String(8) Integer string(4) Text(255) Boolean	(1,0)
PetNote	id_petnote titre description date heure	Identifiant de la note Titre de la note Description de la note Date de création heure de la création	BigInteger Varchar String(255) Date Time	
Alerte	id_alerte type contenu statut	Identifiant de l'alerte Type de l'alerte Contenu État de l'alerte	BigInteger Varchar Text(255) Enum	(en attente, accepté, refusé, supprimé)
Messagerie	id_message emeteur recepteur	Identifiant de messagerie Émetteur Récepteur	BigInteger BigInteger BigInteger	

	Contenu Vu date envoi heur d'envoi	Corps du message Vu Date d'envoi l'heure d'envoi	String(255) Boolean Date Time	(0,1)
Adoption	id_adoption statut date heure N_b	Identifiant de l'adop- tion État de l'adoption Date de l'adoption l'heure de l'adoption remarque	BigInteger enum Date hour text(255)	(en_attente ,accepté, refusé)
Publication	id_publication contenu image vidéo date_creation heure_creation	Identifiant de la publi- cation Contenu Image Vidéo Date de création Heure de création	BigInteger Text Varchar Varchar Date Time	
Like	id_like	Identifiant de j'aime	BigInteger	
Commentaire	id_commentaire contenu date heure	Identifiant de com- mentaire Corps du commen- taire Date de commentaire heure	BigInteger Text Date Time	
RendezVous	id_rendezvous date heure statut note	Identifiant de rendez- vous Date Heure État du rendez-vous Note	BigInteger Date Time Enum Text	(en_attente, acceptée, refusée)
Signalement	id_signalement reason signalement Date	Identifiant de signale- ment reason de signalement date de signalement	BigInteger text(255) Date	

TABLE 6.1 – Dictionnaire de données

6.4.1. Schéma relationnel

Les règles de passage du diagramme de classe vers le modèle relationnel

Pour garantir une représentation précise des données lors de la conversion du diagramme de classe au modèle relationnel, il est essentiel de suivre un ensemble de règles. Voici les directives principales à observer :

1. Classe vers Table : Chaque classe du diagramme de classe est généralement traduite en une table dans le modèle relationnel, avec le même nom.
2. Attributs vers Colonnes : Les attributs de chaque classe sont généralement représentés par des colonnes dans leurs tables respectives, avec le même nom.
3. Clé primaire : L'identification de la clé primaire est cruciale, avec les attributs formant la clé primaire définis comme tels.
4. Relation un-à-un : Les relations un-à-un peuvent être traduites en incluant la clé primaire d'une classe comme clé étrangère dans l'autre classe.
5. Relation un-à-plusieurs : Les relations un-à-plusieurs peuvent être traduites en incluant la clé primaire de la classe « Un » comme clé étrangère dans la classe « plusieurs ».
6. Relation plusieurs-à-plusieurs : Les relations plusieurs-à-plusieurs nécessitent généralement une table de jointure supplémentaire contenant les clés primaires des deux classes.
7. Héritage : Pour traduire une association d'héritage, trois décompositions sont possibles :
 - (a) Décomposition par distinction : Chaque sous-classe devient une relation avec la clé primaire migrée de la surclasse.
 - (b) Décomposition descendante : En cas de contrainte de totalité ou de partition, où les attributs de la surclasse migrent vers les sous-classes.
 - (c) Décomposition ascendante : Supprime la relation issue de la sous-classe et transfère les attributs vers la surclasse.

6.4.2. Modèle relationnel

Pour notre conception, voici le schéma relationnel obtenu, où les données sont organisées en tables selon le modèle relationnel, facilitant la création de nouvelles relations à partir d'opérations sur d'autres tables :

- **Utilisateur** (id_user, nom_utilisateur, email, mot_de_passe, usertype, photo)
- **Propriétaire** (numéro_de_téléphone, adresse, #id_user)
- **Administrateur** (#id_user)
- **Vétérinaire** (nomveto, Prénomveto, Numtél, Horaire_de_travail, NomCabinet, localisation,description, is_approved, #id_user)
- **Adoptant** (NomAdoptant, PrénomAdoptant, numtelp, adressep, #id_user)
- **Petprofil** (id_pet, Nom, Espèce, Race, Age, Sexe, Description, Image, #id_propriétaire)
- **PetNote** (id_PetNote, titre, date_creation, heure_creation, description,#pet-id)
- **Alerte** (id_alerte, type, Date,Contenu_alerte, status, #propriétaire_id)
- **Rendez-vous** (id_rdv, Date, Heure, status_rdv, N_B_rdv, #id_propriétaire, #id_pet, #id_vétérinaire)
- **Message** (id_message, Contenu, emetteur, recepateur, vu, Date_d'envoi, Heure_d'envoi,id_user)
- **Publication** (id_publication, contenu, image, video, date_création, heure_création, #id_user)

- **Commentaire** (id_commentaire, contenue, date_commentaire, heure_commentaire, #id_post, #id_user)
- **Like** (id_like, #id_user, #id_publication)
- **Adoption** (id_adoption, Date_adoption, Heure_adoption, #id_adoptant, #id_propriétaire)
- **Signalement** (id_signal, reason_signalement, Date, #id_adoptant, #id_propriétaire, #id_vétérinaire, #id_post)

6.5. Conclusion

Ce chapitre nous a permis de comprendre l'importance cruciale de la modélisation de la base de données pour notre application. À travers l'utilisation du diagramme de classes, des règles de gestion et du dictionnaire de données, nous avons pu garantir la cohérence et la précision de notre conception. Ces outils ont facilité une structuration efficace des données, permettant ainsi de répondre aux besoins spécifiques de notre application. Dans le chapitre suivant nous présenterons l'environnement de développement ainsi que quelques interfaces de notre application.

Chapitre 7

Réalisation

7.1. Introduction

Dans ce chapitre, nous définirons la réalisation de notre application en mettant en avant l'environnement de développement et les technologies employées pour l'implémentation de notre application. Par la suite, nous présenterons quelques interfaces du système pour illustrer les fonctionnalités implémentées.

7.2. Langages, outils et environnement de développement

7.2.1. Côté Backend

Langages utilisés

PHP (HyperText Preprocessor) est un langage de script côté serveur qui est spécialement utilisé pour le développement web. Il est principalement destiné à la génération de contenu dynamique ainsi qu'à la manipulation des bases de données.¹



FIG 6. : Logo de PHP

Framework utilisés

Laravel est l'un des Framework du langage de programmation PHP le plus populaire. Il est open-source et orienté objet. Créé par Taylor Otwell en 2011, il est destiné au développement des applications web suivant le modèle architectural MVC (Model View Controller). Laravel offre plusieurs fonctionnalités telles que l'authentification, la gestion des sessions ainsi que la manipulation et la migration des bases de données via Eloquent ORM.²



FIG 6. : Logo de Laravel

¹PHP. <https://www.php.net/> (visité le 29 mai 2024).

²Laravel. <https://laravel.com/> (visité le 29 mai 2024).

Base de données utilisée

MySQL (My Structured Query Language) est un SGBDR (système de gestion de base de données relationnelles) open-source. Il est largement conçu pour stocker et manipuler des bases de données. Il utilise le langage SQL (Structured Query Language) qui prend en charge les opérations CRUD (Create, Read, Update, Delete), les quatre fonctions basiques de la gestion des bases de données.³



FIG 6. : Logo de MySQL

7.2.2. Côté Frontend

Langages utilisés

HTML (HyperText Markup Language) est le langage standard utilisé pour créer des pages web. Il permet de structurer le contenu d'une page en utilisant des balises.⁴ CSS (Cascading Style Sheets) est un langage de feuille de style utilisé pour décrire l'apparence et la mise en forme d'un document écrit en HTML. Il permet de contrôler la mise en page d'un site web.⁵



FIG 6. : Logo de HTML5 et CSS 3

JavaScript est un langage de programmation de scripts orienté objet, côté client. Il est principalement utilisé dans les pages web interactives. Il permet de manipuler le DOM (Document Object Model) et de communiquer avec les serveurs via AJAX.⁶



FIG 6. : Logo de JavaScript

³MySQL. <https://fr.wikipedia.org/wiki/MySQL>(visité le 29/05/2024)

⁴HTML. <https://developer.mozilla.org/fr/docs/Web/HTML>(visité le 29/05/2024)

⁵CSS. https://developer.mozilla.org/fr/docs/Learn/CSS_basics(visité le 29/05/2024)

⁶Js. <https://developer.mozilla.org/fr/docs/Web/JavaScript> (visité le 29/05/2024)

Framework utilisés

Bootstrap est un Framework CSS frontend et open-source, utilisé pour le développement. Il offre une collection de composants HTML, CSS et JavaScript qui facilite le développement et la création des designs des sites et des applications web. ⁷



FIG 6. : Logo de Bootstrap

Pusher est un service de messagerie en temps réel qui facilite la création des applications interactives. Il permet des fonctionnalités telles que la mise à jour instantanée des données, les notifications et les chats en direct dans les applications mobiles et web. ⁸



FIG 6. : Logo de Pusher

7.2.3. environnement de développement

Node Js est un environnement d'exécution JavaScript côté serveur, multiplateforme et open-source. Il est largement utilisé pour développer des serveurs, des scripts et des APIs. Grâce à son écosystème de modules npm (gestionnaire de paquets), Node js est l'un des choix les plus populaires pour les développeurs. ⁹



FIG 6. : Logo de NodeJs

VS Code (Visual Studio Code) :est un éditeur de code source open-source développé par Microsoft. Connue pour sa flexibilité et sa rapidité, il offre un environnement de développement riche avec des outils de débogage intégrés et une vaste bibliothèque d'extensions pour divers langages de programmation. ¹⁰

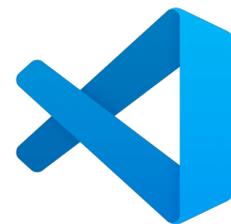


FIG 6. : Logo de Vscode

⁷Bootstrap. <https://getbootstrap.com/> (visité le 29/05/2024)

⁸Pusher. <https://pusher.com/laravel/> (visité le 29/05/2024)

⁹NodeJs. <https://nodejs.org/en/> (visité le 29/05/2024)

¹⁰Vscode. <https://code.visualstudio.com/> (visité le 29/05/2024)

Github : est une plateforme de développement collaborative reposant sur Git, qui permet aux développeurs de collaborer et de contribuer à des projets partagés tout en gardant un suivi détaillé de leur progression. Facilitant ainsi le travail en équipe et la contribution à des projets open-source tout en offrant des outils pour le contrôle de version, la gestion des branches et la gestion des conflits.¹¹



FIG 6. : Logo de github

Figma : est un outil de design graphique et prototypage largement utilisé pour la conception des site web et d'interfaces d'applications. Il est principalement basé sur le Cloud, offrant une collaboration en temps réel et différentes fonctionnalités avancées telles que la création de maquettes et de prototypes interactifs ainsi que le partage de fichiers.¹²

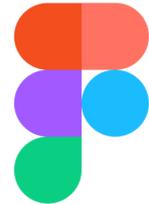


FIG 6. : Logo de figma

Lucidchart est un outil en ligne pour créer et visualiser des diagrammes comme les flux de travail, les organigrammes, et les cartes mentales. Il offre une interface intuitive, des options de collaboration en temps réel, et s'intègre à divers outils pour faciliter le travail d'équipe et le partage des visuels.¹³



FIG 6. : Logo de lucidchart

¹¹Github. <https://github.com/> (visité le 29/05/2024)

¹²Figma. <https://www.figma.com/fr-fr/> (visité le 29/05/2024)

¹³Lucidchart. <https://lucid.app/>

7.3. Etude comparative finale

Nous avons repris l'étude comparative effectuée dans le premier chapitre (Contexte d'étude et recueil des besoins page 8) en ajoutant la comparaison entre les deux applications Wamiz et Yummypets avec notre application Pettracker.

Fonctionnalité	Wamiz	Yummypets	Petracker
Espace de publication	✓	✓	✓
Espace d'alerte d'urgence	✗	✗	✓
Espace Messagerie	✓	✗	✓
Espace rendez-vous	✗	✗	✓
Espace Adoption	✓	✓	✓
Espace de création des comptes animaux	✗	✓	✓
Mini agenda	✗	✗	✓
Facilité d'utilisation	Bonne	Bonne	Bonne
Disponibilité (web/mobile)	Web et mobile	Web et mobile	Web uniquement

TABLE 7.1 – Comparaison des fonctionnalités des applications

7.3.1. Analyse comparative

- **Wamiz** : Propose des fonctionnalités de publication, messagerie et adoption, mais manque d'alertes d'urgence, de gestion des rendez-vous, de création de comptes animaux et de mini agenda. Disponible sur web et mobile.
- **Yummypets** : Offre des fonctionnalités de publication, d'adoption et de création de comptes animaux, mais ne propose pas d'alertes d'urgence, de messagerie entre utilisateurs, de gestion des rendez-vous ni de mini agenda. Disponible sur web et mobile.
- **Petracker** : Fournit une gamme complète de fonctionnalités incluant les alertes d'urgence, la gestion des rendez-vous, la création de comptes animaux et un mini agenda, en plus des fonctionnalités de base. Disponible uniquement sur le web.

7.4. Présentation des interfaces

Dans ce qui suit, nous présentons quelques interfaces de notre application « Petracker ».

7.4.1. Interface d'accueil

La figure suivante représente l'interface d'accueil de notre plateforme « petracker ».

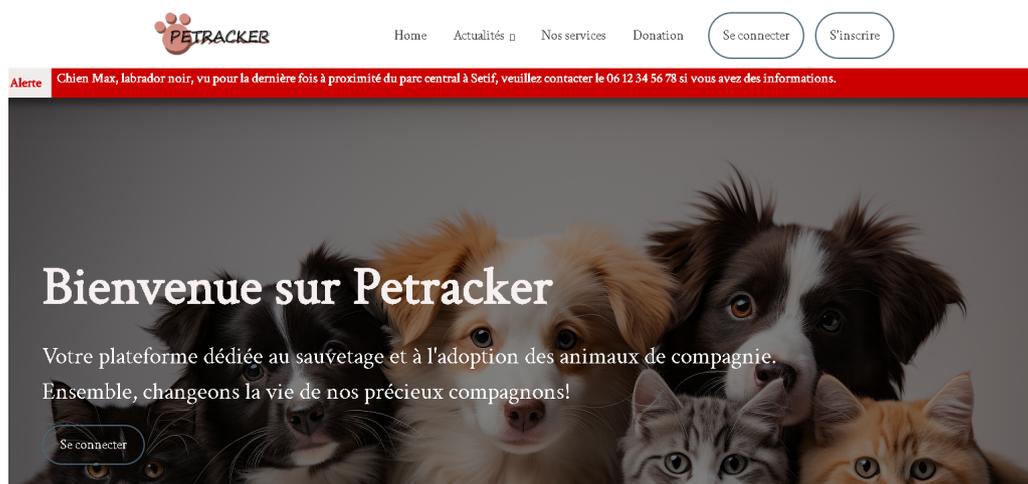


fig 7.1 : Interface d'accueil de petracker (partie 1)

7.4.2. Interface d'accueil

La figure suivante présente l'interface d'accueil de notre plateforme « petracker partie 02 ».

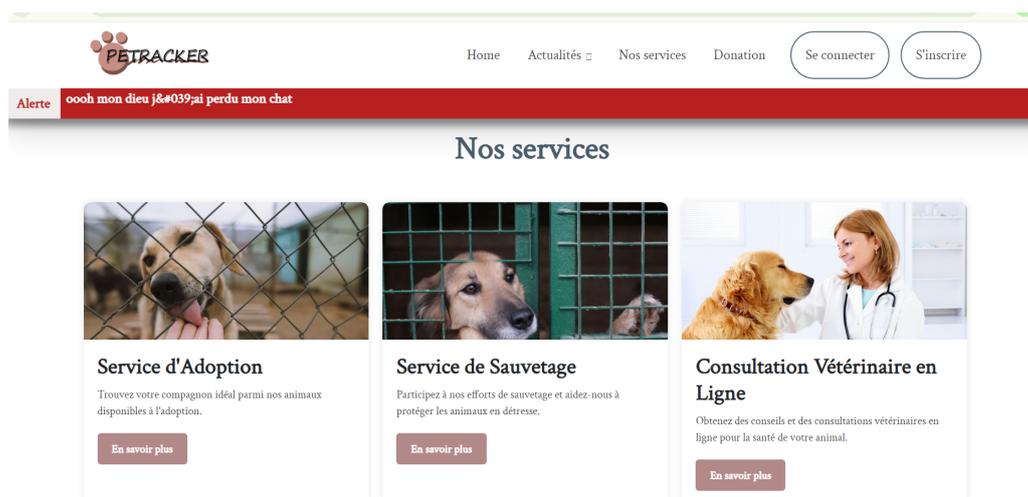
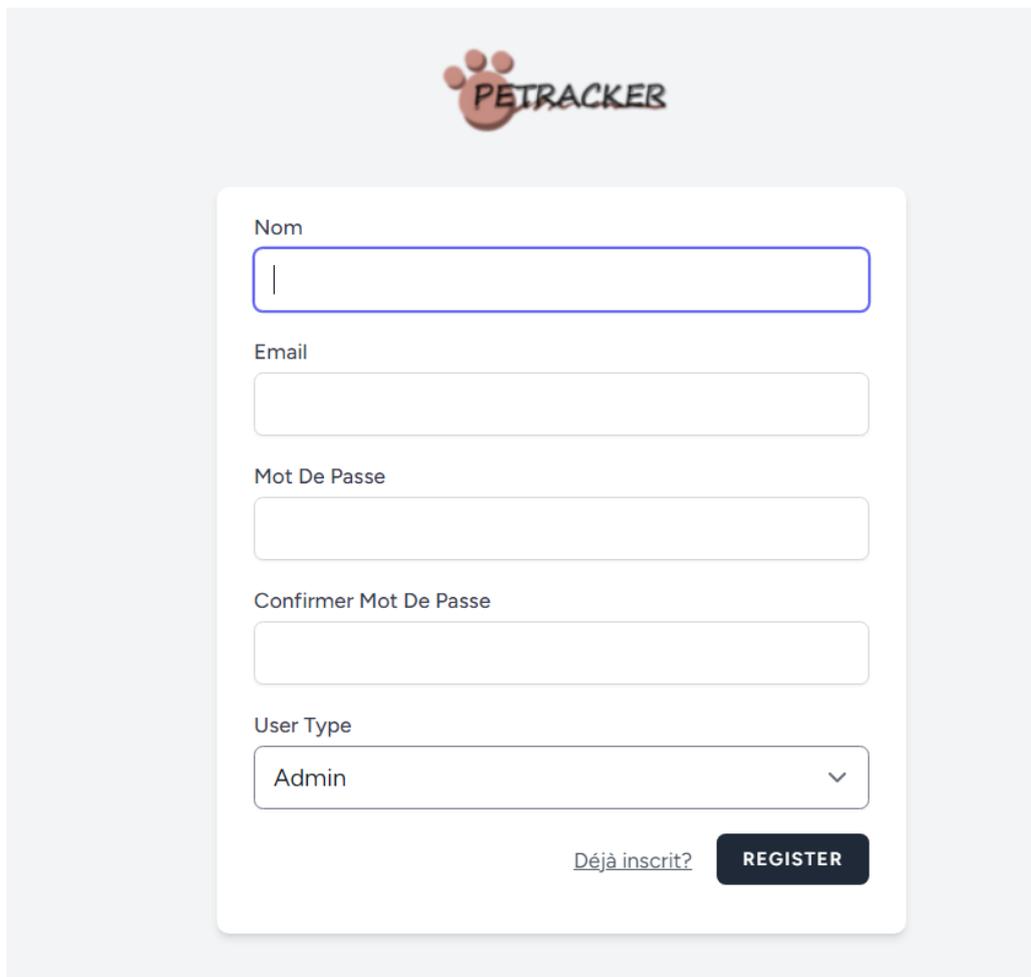


fig 7.2 : Interface d'accueil de petracker (partie 2)

7.4.3. Interface d'inscription

La figure suivante représente l'interface d'inscription de notre plateforme « petracker ».



The image shows a registration form for a platform named 'PETRACKER'. At the top center is the logo, which consists of a red paw print with the word 'PETRACKER' in a bold, black, sans-serif font. Below the logo is a white rectangular form with rounded corners. The form contains the following fields from top to bottom: a text input field labeled 'Nom' with a vertical cursor; a text input field labeled 'Email'; a text input field labeled 'Mot De Passe'; a text input field labeled 'Confirmer Mot De Passe'; and a dropdown menu labeled 'User Type' with 'Admin' selected and a downward arrow. At the bottom right of the form, there is a link labeled 'Déjà inscrit?' and a dark blue button with the word 'REGISTER' in white capital letters.

fig 7.3 : Interface d'inscription de petracker

7.4.4. Interface de connexion

La figure suivante présente l'interface de connexion de notre plateforme « petracker ».

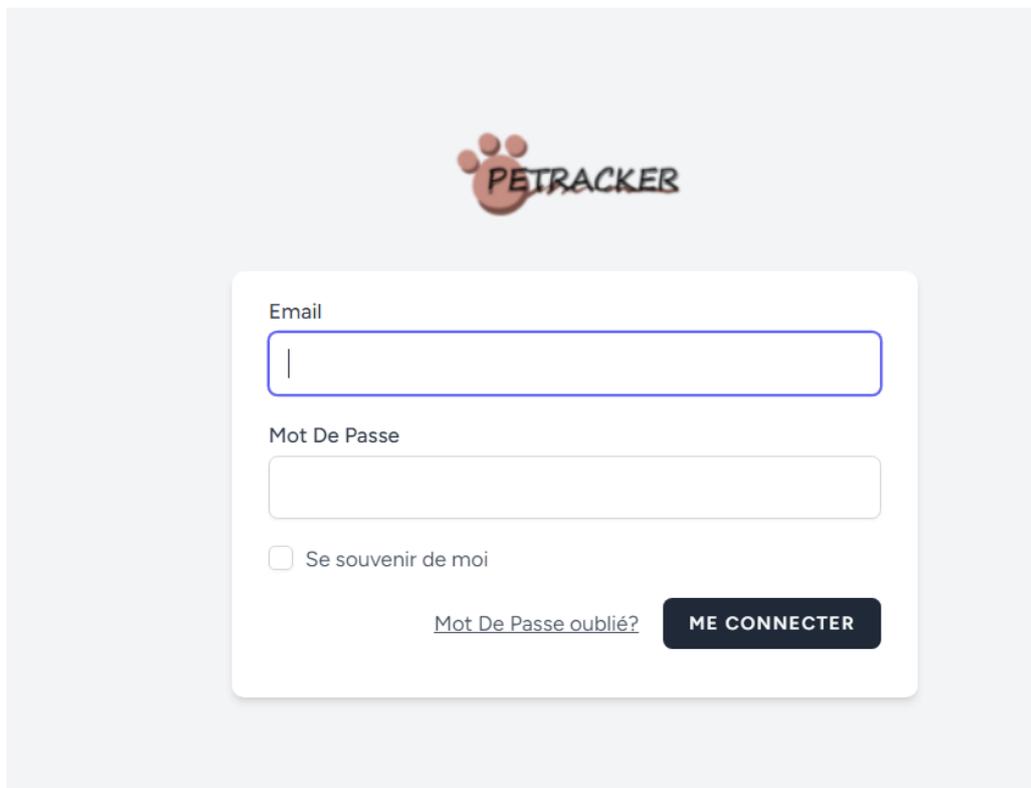


fig 7.4 : interface de connexion

7.4.5. Interface Espace propriétaire

La figure suivante représente l'interface de l'espace propriétaire de notre plateforme « pet-tracker ».

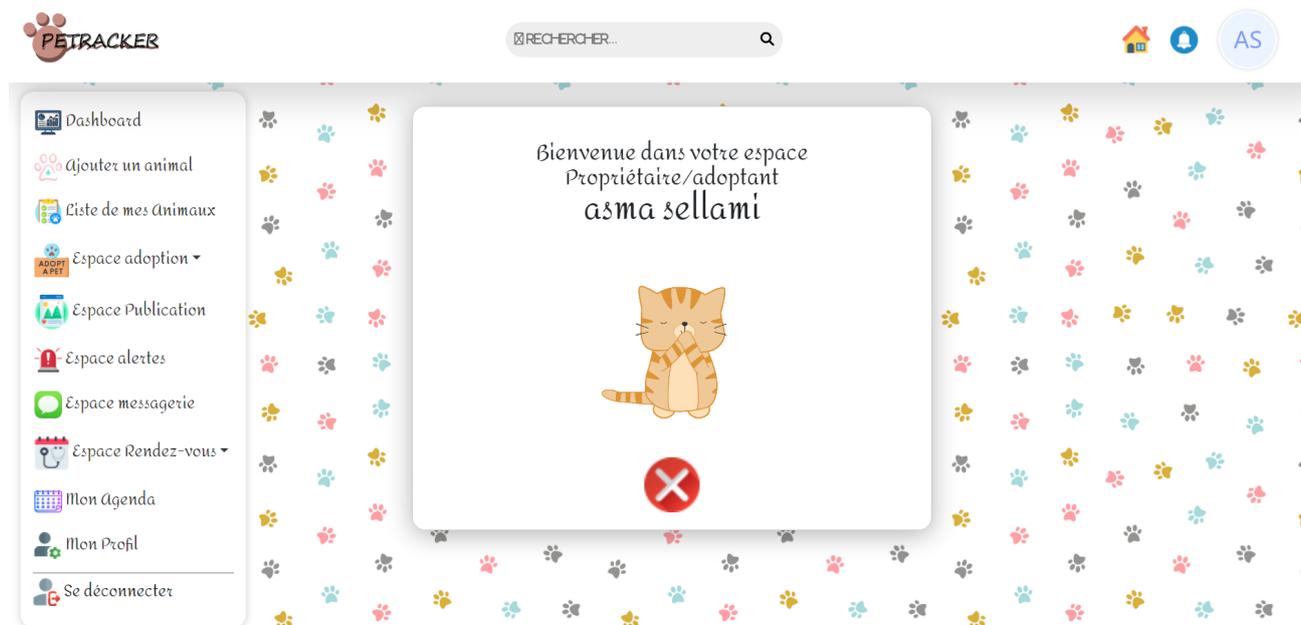


fig 7.5 : Interface d'accueil de l'espace propriétaire

7.4.6. Interface creation compte animal

La figure suivante représente la première partie de l'interface de création d'un compte animal,

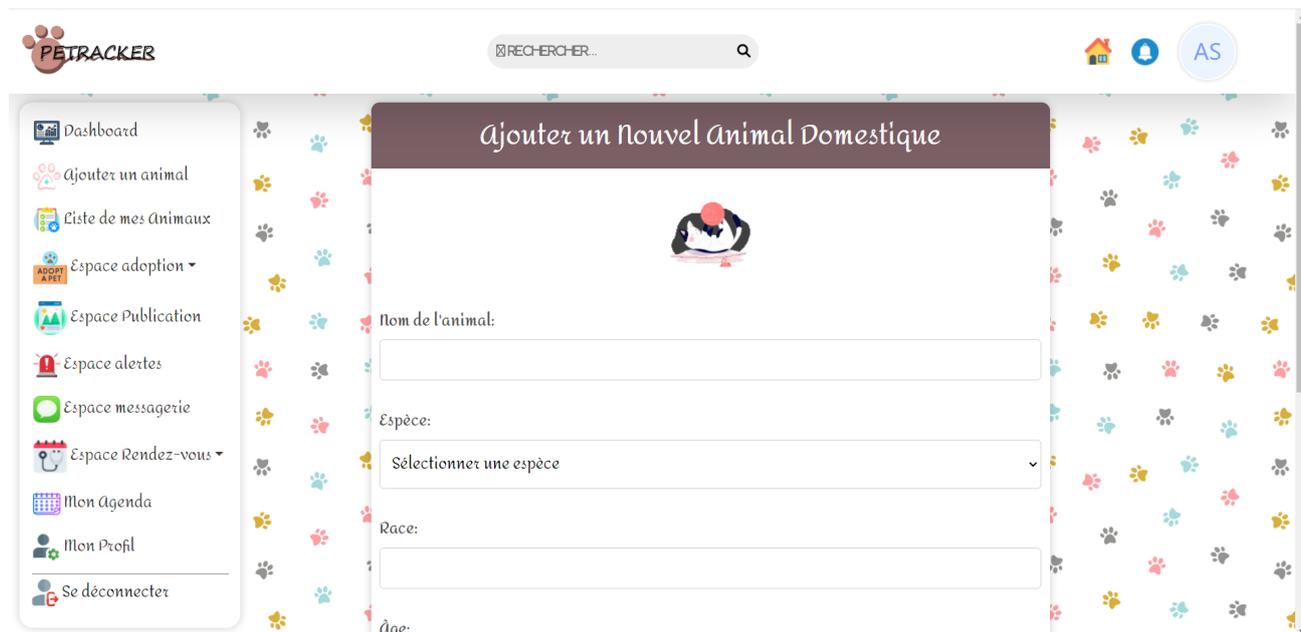


fig 7.6 : Interface de création d'un compte animal partie 1

7.4.7. Interface Espace création compte animal

La figure suivante représente la deuxième partie de l'interface créer compte animal de notre plateforme « pettracker ».

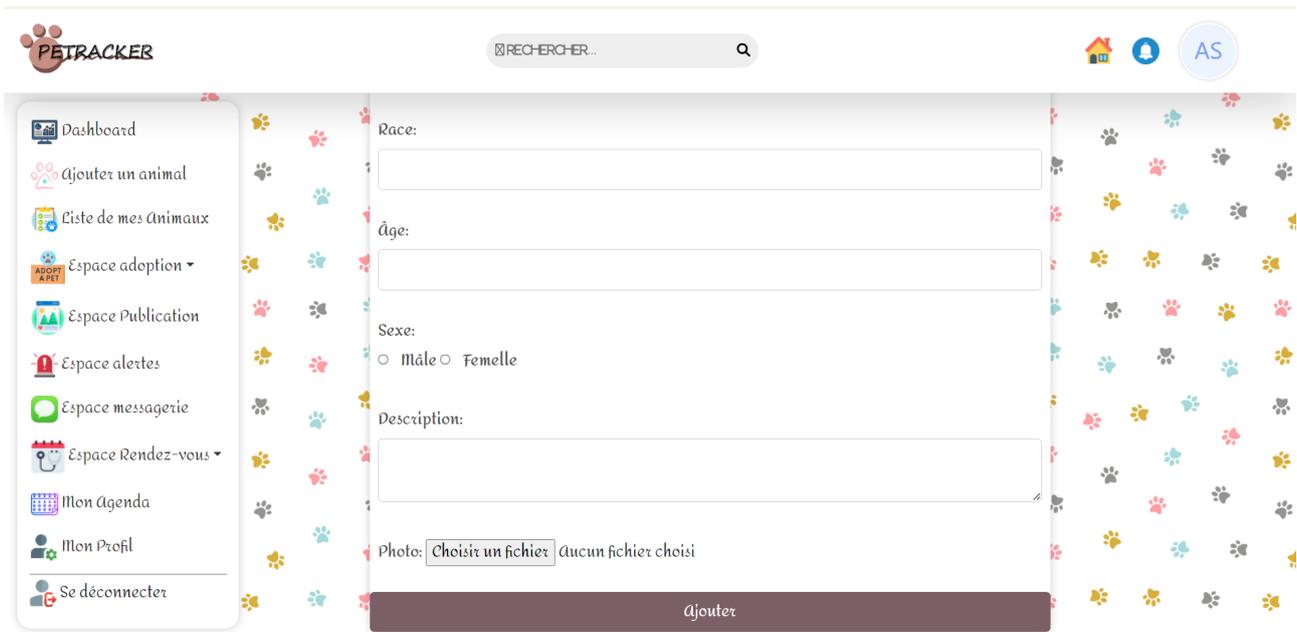


fig 7.7 : Interface de creation d'un compte animal partie2

7.4.8. Interface espace liste de mes animaux

La figure suivante représente l'interface liste des animaux de notre plateforme « petracker ».

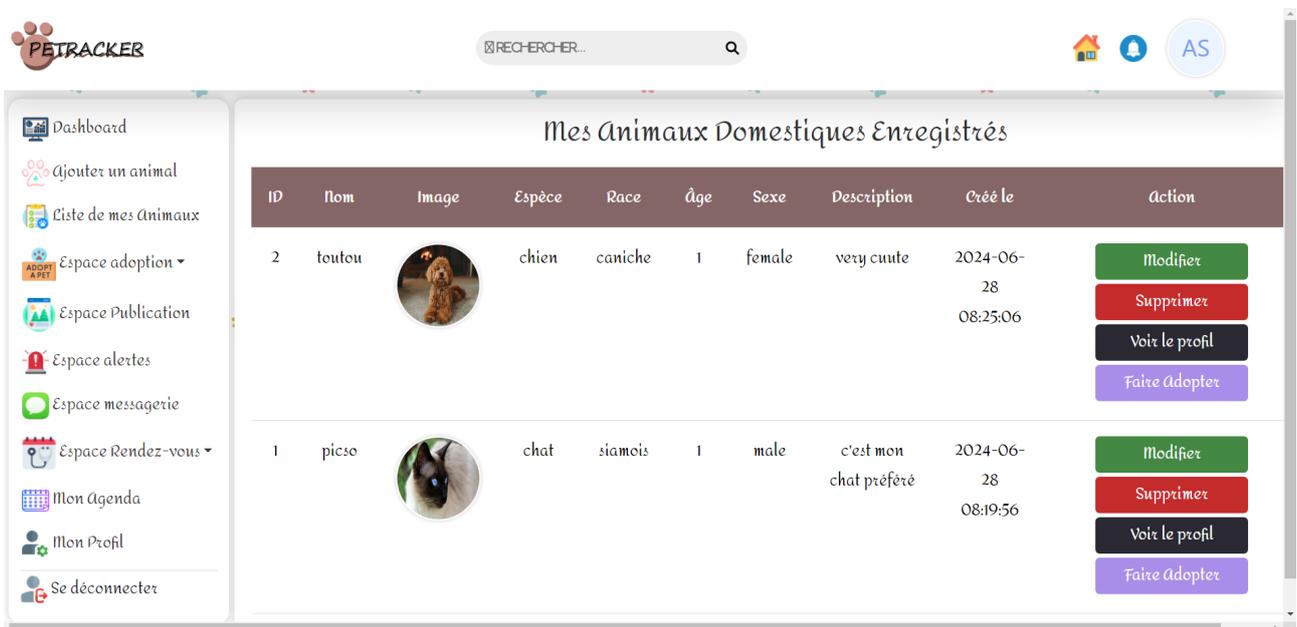


fig 7.8 : Interface liste de mes animaux

7.4.9. Interface profil d'un animal

La figure suivante représente l'interface du profil animal de notre plateforme « petracker ».

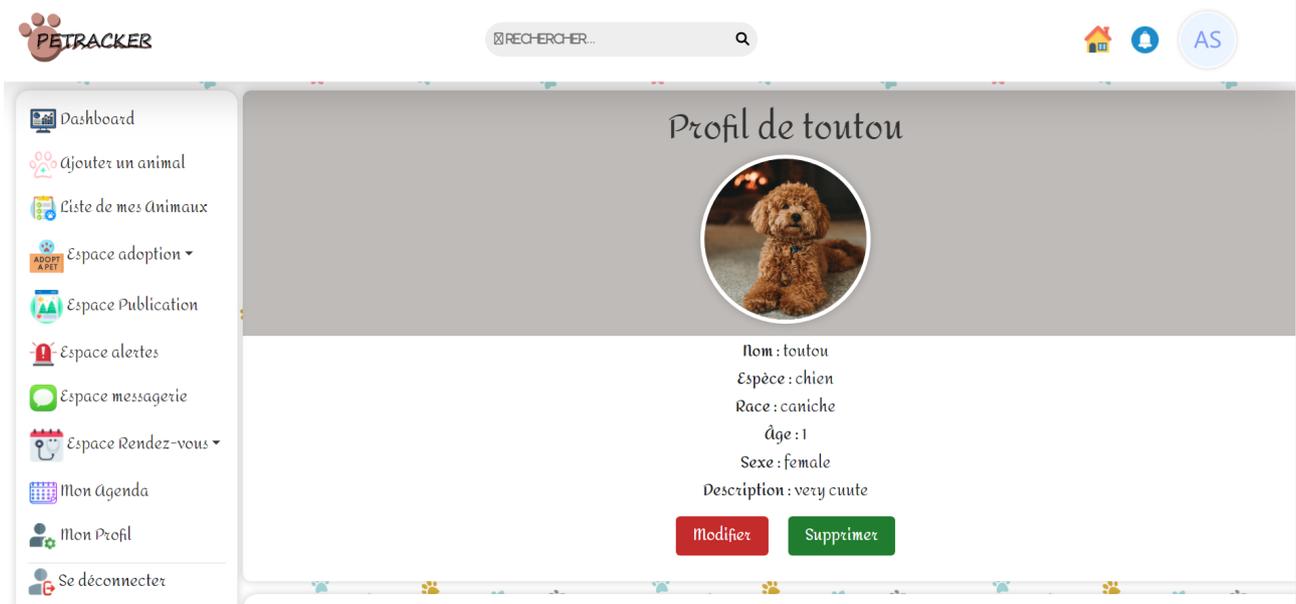


fig 7.9 : Interface profil animal

7.4.10. Interface de la prise de note

La figure suivante représente l'interface de prise de note d'un animal de notre plateforme « petracker ».

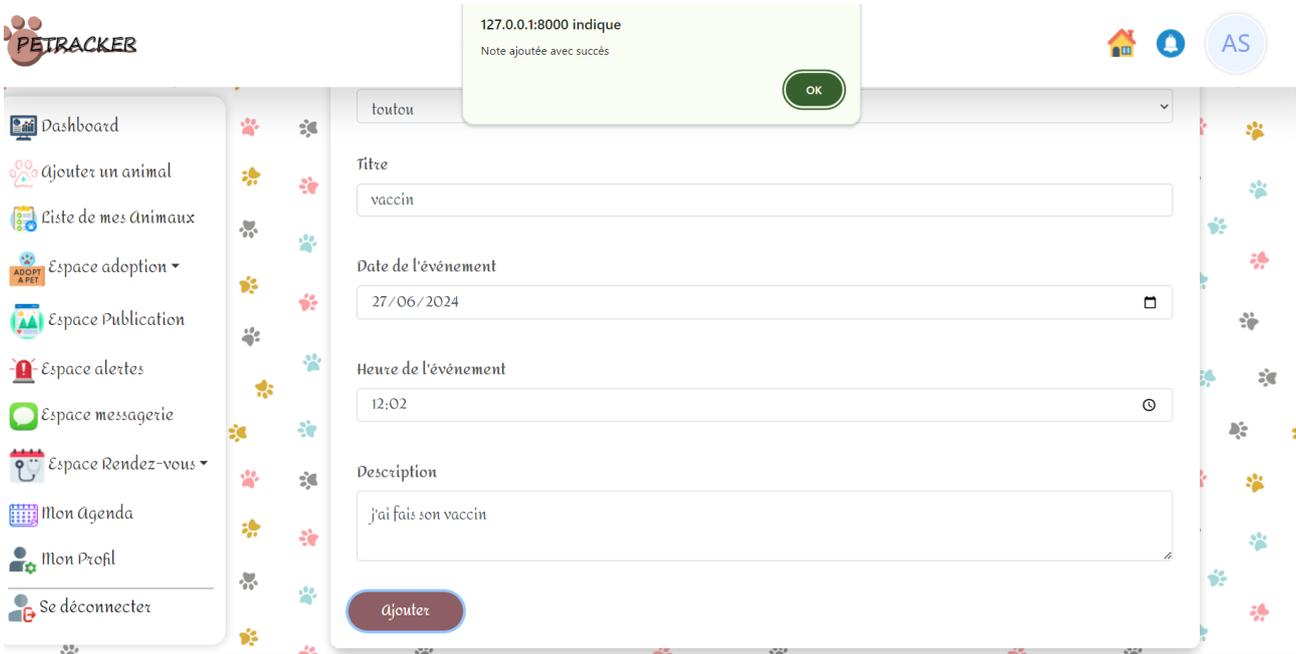


fig 7.10 : Interface prise note pour animal

7.4.11. Interface de la liste des notes prises

La figure suivante représente l'interface de la liste des notes prises pour un animal dans notre plateforme « petracker ».



fig 7.11 : Interface note pour animal

7.4.12. Interface Espace publication

La figure suivante représente l'interface publier de notre plateforme « petracker ».

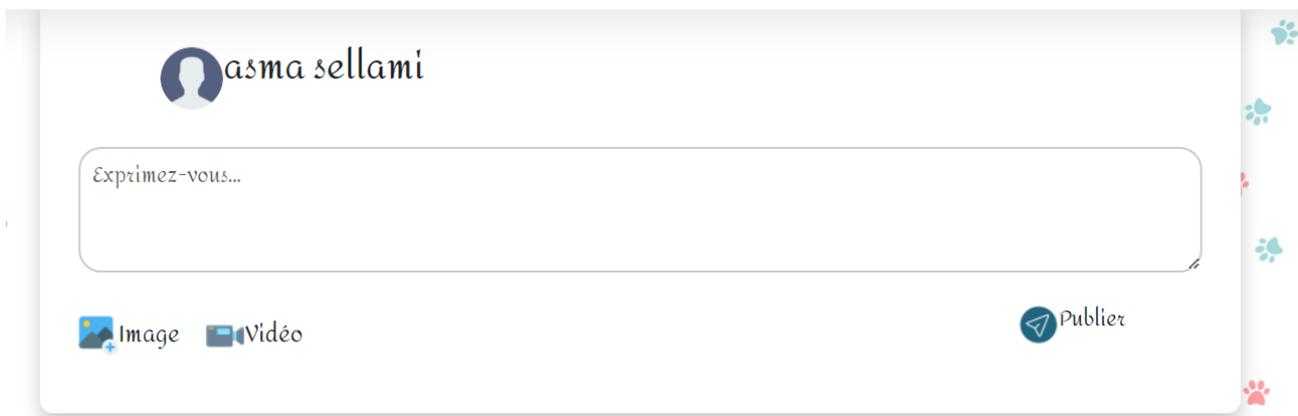


fig 7.12 : Interface publier

7.4.13. Interface publication

La figure suivante représente l'interface publication de notre plateforme « petracker ».

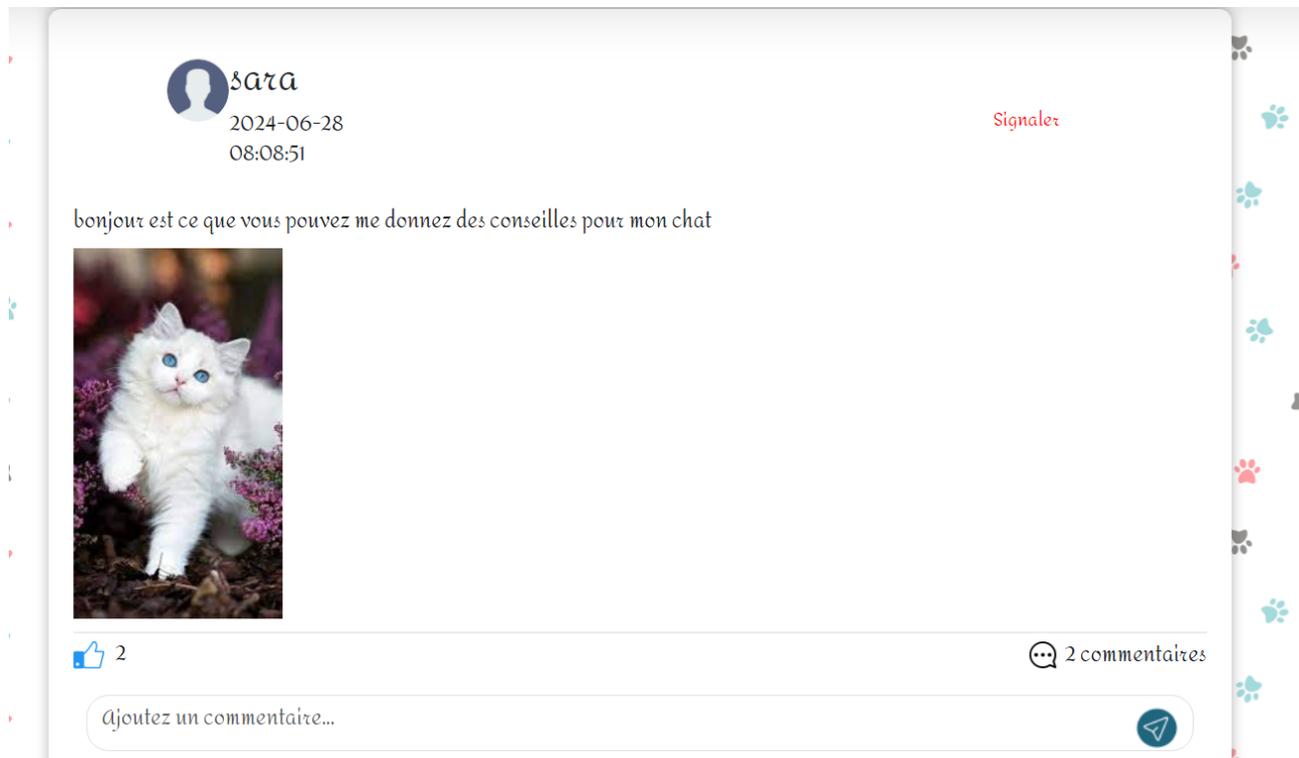


fig 7.13 : Interface publication

7.4.14. Interface signaler une publication

La figure suivante représente le signalement des publications de notre plateforme « petracker ».

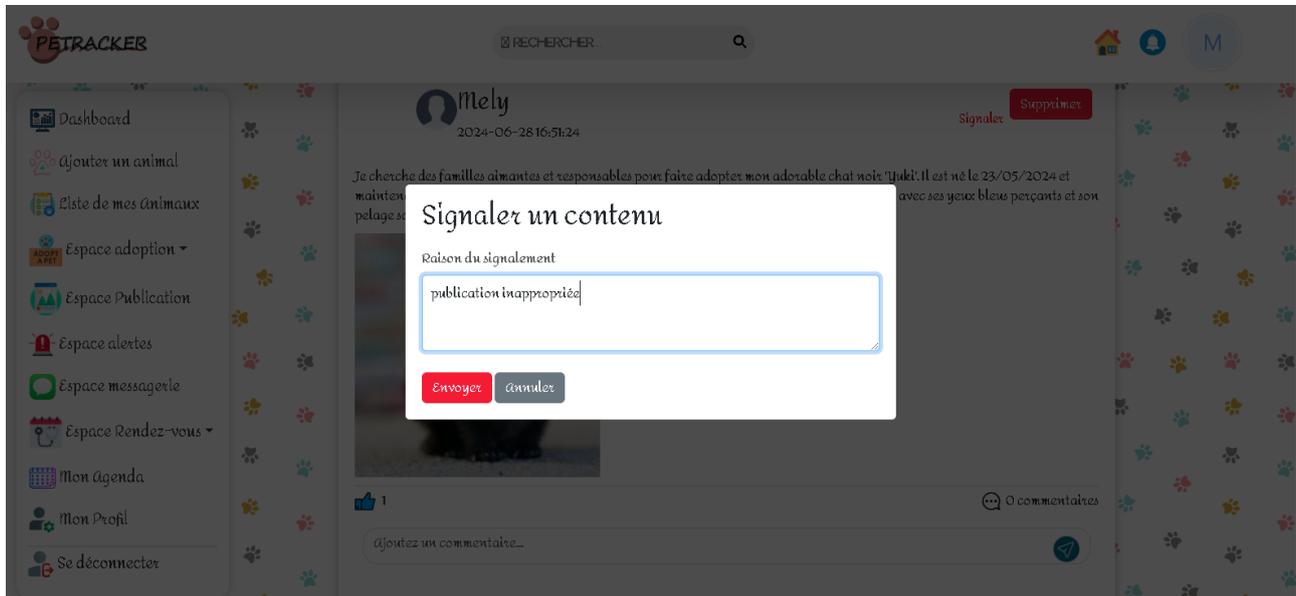
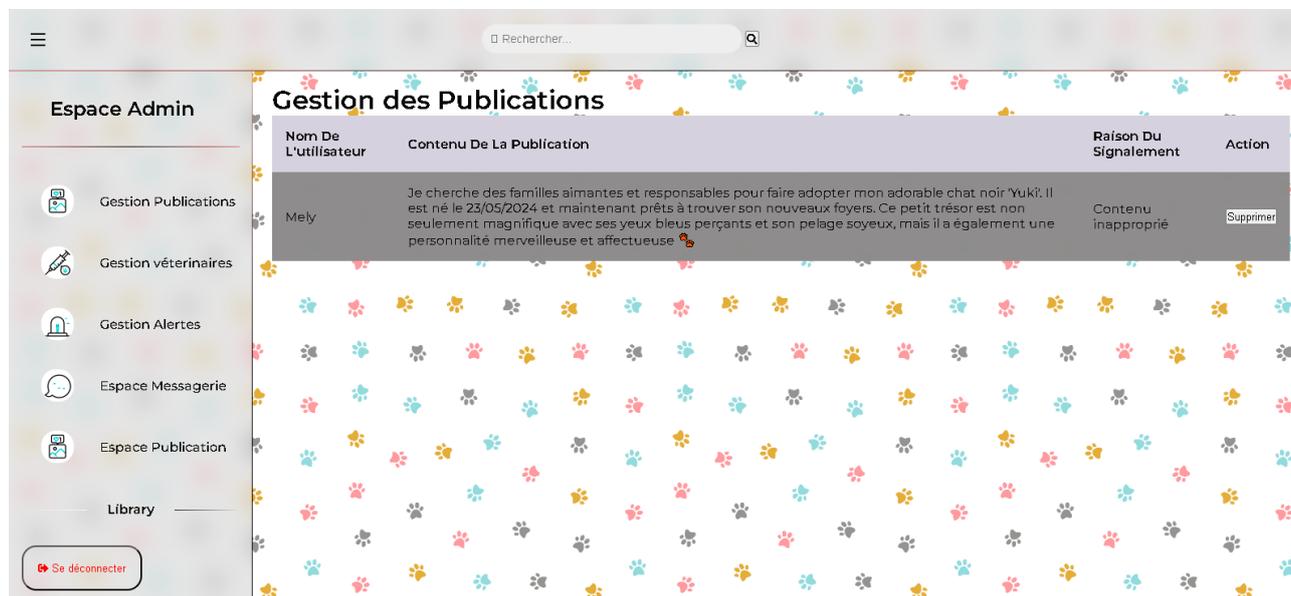


fig 7.14 : signaler la publication

7.4.15. Interface de gestion des signalement des publications

La figure suivante représente la liste des signalements des publications qui sera gérée par l'administrateur de notre plateforme « petracker ».



Nom De L'utilisateur	Contenu De La Publication	Raison Du Signalement	Action
Mely	Je cherche des familles aimantes et responsables pour faire adopter mon adorable chat noir 'Yuki'. Il est né le 23/05/2024 et maintenant prêts à trouver son nouveaux foyers. Ce petit trésor est non seulement magnifique avec ses yeux bleus perçants et son pelage soyeux, mais il a également une personnalité merveilleuse et affectueuse.	Contenu inapproprié	Supprimer

fig 7.15 : signaler la publication

7.4.16. Interface messagerie

La figure suivante représente l'interface messagerie de notre plateforme « petracker ».

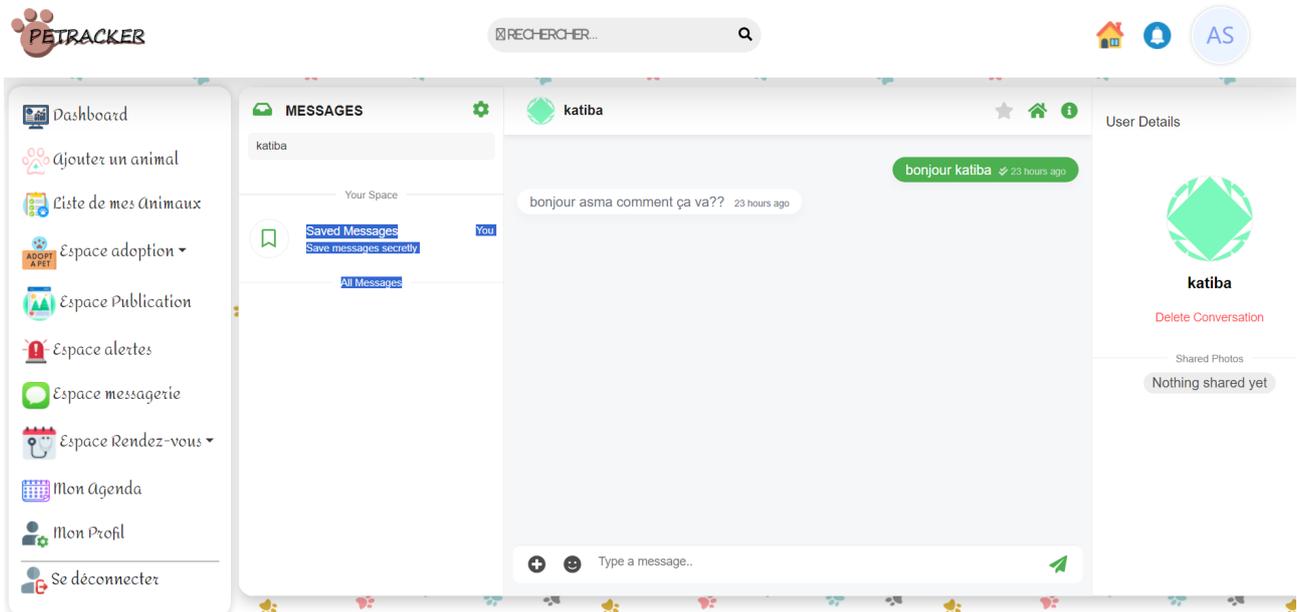


fig 7.16 : Interface messagerie

7.4.17. Interface liste des vétérinaire

La figure suivante représente la liste des vétérinaires de notre plateforme « petracker ».



fig 7.17 : Interface liste vétérinaire

7.4.18. Interface profil du vétérinaire

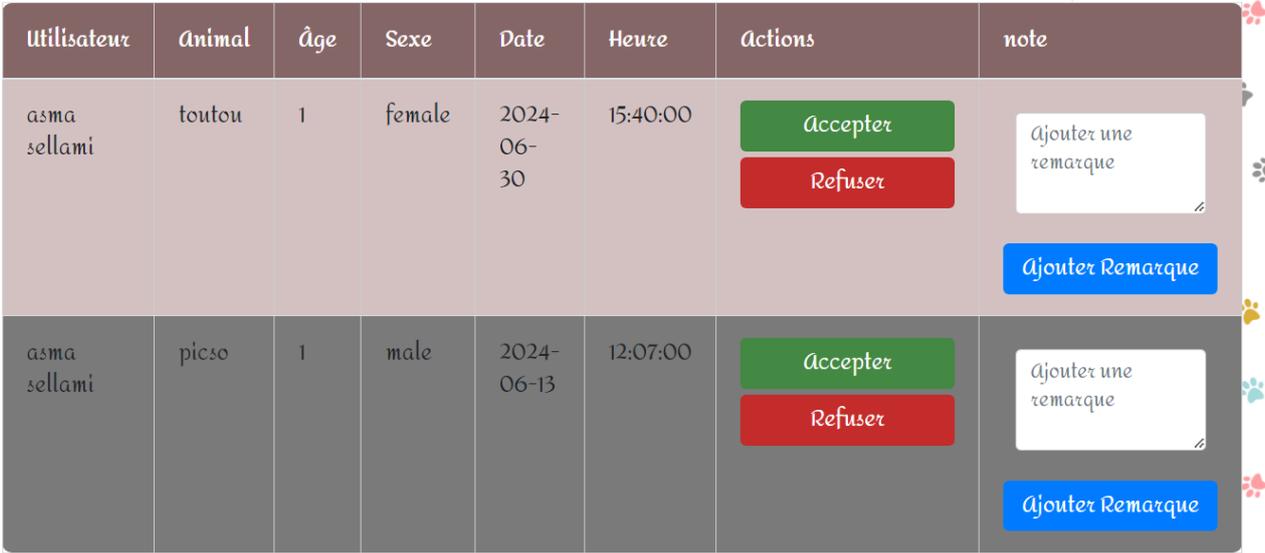
La figure suivante représente l'interface le profilde vétérinaires de notre plateforme « petra-cker ».



fig 7.18 : Interface profil du vétérinaire

7.4.19. Interface de la liste des demandes de rendez-vous

La figure suivante représente la liste des demande de rendez-vous qui sera géré par le vétérinaire



Utilisateur	Animal	Âge	Sexe	Date	Heure	Actions	note
asma sellami	toutou	1	female	2024-06-30	15:40:00	<input type="button" value="Accepter"/> <input type="button" value="Refuser"/>	<input type="text" value="Ajouter une remarque"/> <input type="button" value="Ajouter Remarque"/>
asma sellami	picso	1	male	2024-06-13	12:07:00	<input type="button" value="Accepter"/> <input type="button" value="Refuser"/>	<input type="text" value="Ajouter une remarque"/> <input type="button" value="Ajouter Remarque"/>

fig 7.19 : Interface liste des rendez-vous

7.4.20. Interface prise de rendez-vous

La figure suivante représente l'interface prise de rendez-vous de notre plateforme « petracker ».

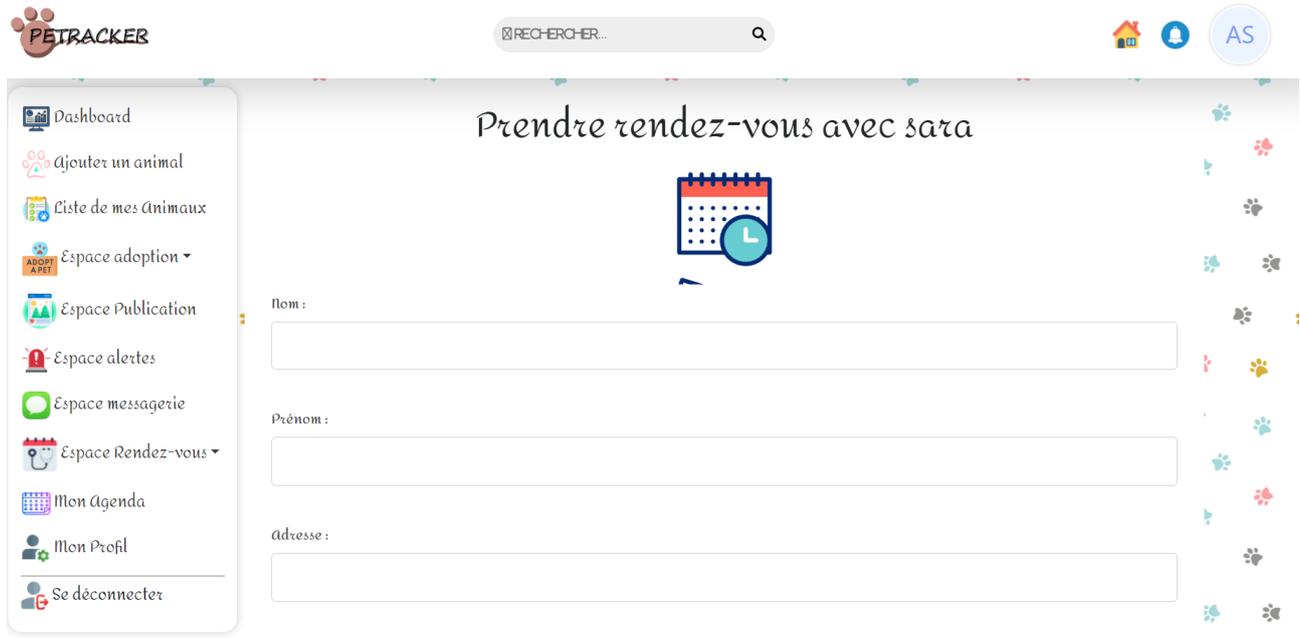


fig 7.20 : Interface prise de rendez-vous

7.4.21. Interface Espace adoption

La figure suivante représente l'interface adoption de notre plateforme « pettracker ».

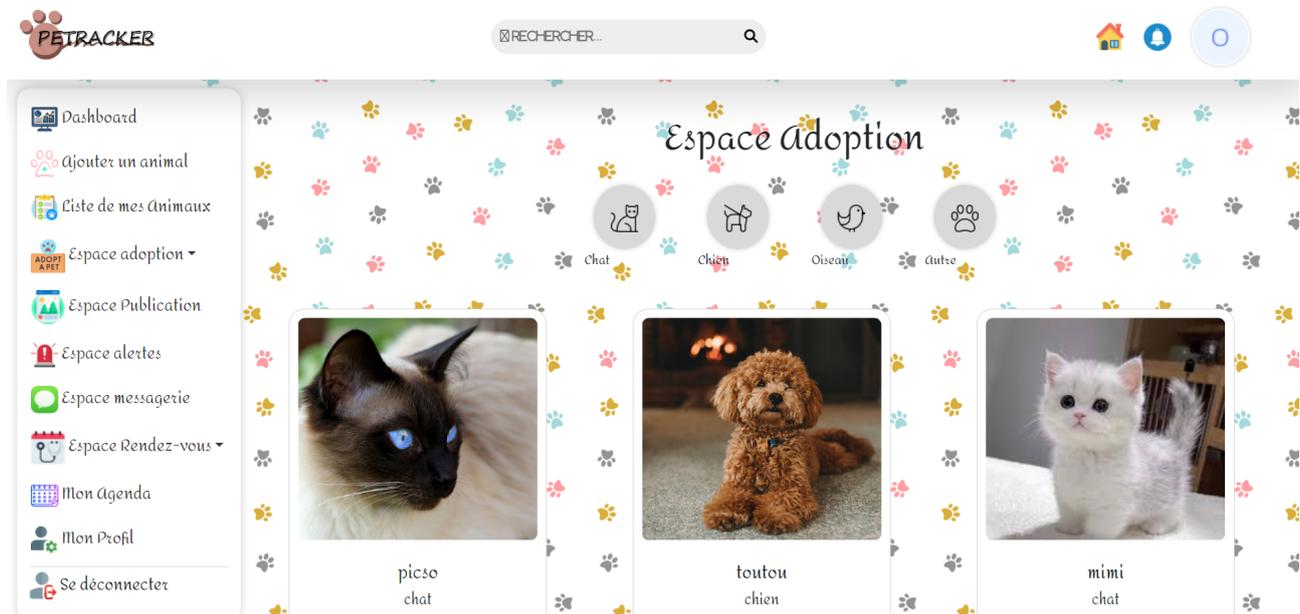


fig 7.21 : Interface adoption

7.4.22. Interface adopter un animal

La figure suivante représente l'interface adopter un animal de notre plateforme « petracker ».

The screenshot shows a web browser window with the URL `:8000/adopt#`. A modal form titled "Adopter un animal" is centered on the screen. The form has three input fields: "Nom complet" with the value "asma", "numéro de téléphone" with the value "0798617783", and "Adresse" with the value "ighil ouazoug". Below the fields are two buttons: a blue "Submit" button and a grey "Annuler" button. Above the form, a light green notification box contains the text "127.0.0.1:8000 indique votre demande a ete envoye avec succes ." and an "OK" button. The background of the page is dark grey with a pattern of colorful paw prints and images of animals, including a Siamese cat and a dog.

fig 7.22 : Interface adopter un animal

7.4.23. Interface de gestion des adoptions

La figure suivante représente la liste des demandes d'adoption qui sera géré par le propriétaire de l'animal .



fig 7.23 : Interface gestion des demandes d'adoptions

7.4.24. Interface alerte

La figure suivante représente l'interface alerte qui permet au propriétaire de signaler des alertes d'urgence .



fig 7.24 : Interface alerte

7.4.25. Interface de gestion des alertes

La figure suivante représente la liste des alertes qui sera géré par l'administrateur .

The screenshot displays the 'Gestion des Alertes' interface. On the left is a sidebar titled 'Espace Admin' with navigation links: 'Gestion Publications', 'Gestion vétérinaires', 'Gestion Alertes', 'Espace Messagerie', 'Espace Publication', and 'Library'. A 'Se déconnecter' button is at the bottom of the sidebar. The main content area features a search bar and the title 'Gestion des Alertes'. Below the title is a table with the following data:

ID	Utilisateur	Type	Message	Status	Actions
4	Asma	perte	Chien Max, labrador noir, vu pour la dernière fois à proximité du parc central à Setif, veuillez contacter le 06 12 34 56 78 si vous avez des informations.	en_attente	Accepter Refuser Supprimer
2	Mely	perte	Chat noir portant un collier bleu, perdu depuis hier les environs de Naceria à Bejaia	acceptée	Supprimer

fig 7.25 : Interface gestion des alertes

7.4.26. Interface demande d'ouverture d'un compte vétérinaire

La figure suivante représente le formulaire de demande d'ouverture d'un compte vétérinaire

The image shows a web form titled "Demande d'ouverture d'un compte vétérinaire". The form is set against a light brown background with a subtle pattern of circles and lines. It contains the following elements:

- Titre:** "Demande d'ouverture d'un compte vétérinaire" in bold black text.
- Champs de saisie:**
 - Nom: Input field with placeholder "Nom".
 - Prénom: Input field with placeholder "Prénom".
 - Numéro du téléphone: Input field with placeholder "Numéro du téléphone".
 - Nom du cabinet: Input field with placeholder "Nom du cabinet".
 - Horraire de travail: Input field with placeholder "Horraire de travail".
 - Frais de la consultation: Input field with placeholder "Frais de la consultation en DA".
 - Localisation: Input field with placeholder "Localisation".
 - Description: Input field with placeholder "Ajoutez une description ici...".
- Photo:** A section with a "Choisir un fichier" button and the text "Aucun fichier choisi".
- Boutons:**
 - A large, light brown button labeled "Envoyer".
 - A dark brown button labeled "Se déconnecter".

fig 7.26 : formulaire de demande d'ouverture d'un compte vétérinaire

7.4.27. Interface de gestion des demandes d'ouverture de comptes vétérinaire

La figure suivante représente la liste des demandes d'ouverture d'un compte vétérinaire qui sera géré par l'administrateur .

Nom	prenom	Numéro de téléphone	Nom du cabinet	Heures de travail	Localisation	Description	Image	Action
oussama	berkani	0565892345	soin pet	8Hjusqua15h	dzaria,alger	je soigne que les chats et les chient		Approuver Supprimer
rania	slimani	0799633556	bimbo	8Hjusqua14h	alger	je soigne avec mon coeur		Approuver Supprimer

fig 7.27 : Interface gestion des demandes d'ouverture de compte vétérinaire

7.5. Conclusion

Dans ce dernier chapitre consacré aux réalisations, nous avons détaillé les outils et technologies utilisés pour développer notre application. Nous avons adopté des langages modernes tels que Laravel et php, permettant une création flexible et dynamique de nos fonctionnalités. L'environnement de développement a été soigneusement choisi pour optimiser la collaboration et l'efficacité, utilisant des plateformes comme Visual Studio Code et GitHub pour la gestion de version et le contrôle du code.

En ce qui concerne les interfaces, nous avons conçu une interface utilisateur intuitive et conviviale. Parmi les exemples présentés, l'écran des espaces de l'ajout d'un profil animal, les espaces d'adoptions expliqué en détail, les espaces de prise de rendez-vous ainsi que l'espace de signalement des alertes d'urgence. Ces réalisations témoignent de notre engagement à fournir une application performante et évolutive, posant ainsi les bases solides pour les développements futurs et l'amélioration continue de notre produit.

Chapitre 8

Conclusion générale

Le développement de Petracker a été une aventure passionnante, au cours de laquelle nous avons pu mettre en pratique nos compétences en génie logiciel et notre expertise dans les technologies web. Notre objectif était ambitieux : créer une plateforme novatrice dédiée à l'adoption et au sauvetage des animaux domestiques, et nous sommes fiers du résultat obtenu. Durant le processus de développement, nous avons été confrontés à divers défis et obstacles. Cependant, ces défis nous ont permis de nous améliorer et d'affiner notre approche. En utilisant des méthodologies de développement telles que la méthode Scrum et le langage de modélisation unifié (UML), nous avons pu structurer notre projet de manière efficace et cohérente. Petracker offre une vaste gamme de fonctionnalités, comprenant la gestion des profils d'animaux, un espace complet dédié à l'adoption des animaux domestiques, un espace de publication pour le partage d'informations au sein de la communauté Petracker, la prise de rendez-vous entre les propriétaires d'animaux et les vétérinaires, ainsi qu'une messagerie intégrée pour tous les utilisateurs de notre plateforme. De plus, nous avons mis en place un système de signalement d'urgences pour une réaction rapide en cas de besoin. Toutes ces fonctionnalités visent à améliorer le bien-être des animaux domestiques et à simplifier la vie de leurs propriétaires.

En conclusion, Petracker représente une avancée significative dans le domaine des technologies dédiées aux animaux domestiques. Notre engagement envers l'innovation continue et l'amélioration constante garantit que Petracker restera à la pointe de son secteur, offrant des solutions pratiques et efficaces pour le bien-être de nos précieux compagnons à quatre pattes. Bien que nous soyons satisfaits des progrès réalisés jusqu'à présent, nous sommes conscients qu'il reste des marges d'amélioration. Dans les prochaines versions de Petracker, nous prévoyons d'intégrer des fonctionnalités telles qu'un espace de dons, une animalerie, une application mobile complémentaire et des améliorations de performance pour une expérience utilisateur optimale.

Chapitre 9

Bibliographie

Toutes les références ont été consultées le 5 mai 2024.

- 1 : Mozilla Developer Network. "Pages, Sites, Servers, and Search Engines." Mozilla Developer Network, 2024. https://developer.mozilla.org/fr/docs/Learn/Common_questions/Web_mechanics/Pages_sites_servers_and_search_engines.
- 2 : Vanilla Academy. Fondamentaux du web : partie statique et dynamique. <https://vanillacademy.com/chapitre/4>
- 3 : SCENARI Community. (n.d.). Schéma d'architecture. Retrieved from <https://stph.scenari-community.org/bdd/lap2/co/webUC003archi.html>
- 4 : Algérie360. Galoufa ou quand les chiens et les chats sont en danger à Alger. <https://www.algerie360.com/galoufa-ou-quand-les-chiens-et-les-chats-sont-en-danger-a-alger>
- 5 : YouTube. <https://youtu.be/Sxbh3rExUcU?si=lvv0iLB-B-oom0ee>.
- 6 : Yummypets. <https://fr.yummypets.com/> .
- 7 : Wamiz. <https://wamiz.com/adoption/>.
- 8 : Wrike. "Qu'est-ce que la méthodologie agile en gestion de projet?". Disponible sur : <https://www.wrike.com/fr/quest-ce-que-la-methodologie-agile-en-gestion-de-projet/>.
- 9 : MEZIANI, Keltoum, et CHALAL, Rosa. Conception et réalisation d'une application web pour la gestion des ordonnances et du stock des médicaments à titre ambulatoire. Mémoire, 2024.

10 : Nutcache. Disponible sur : <https://www.nutcache.com/fr/blog/les-methodes-agiles/>

11 : l'agiliste. <https://agiliste.fr/introduction-methodes-agiles/>

12 : QRP International. <https://www.qrpinternational.fr/glossaire/scrum-cest-quoi-definit>

13 : BLOG Gestion projet. <https://blog-gestion-de-projet.com/>

14 : ORP. <https://www.qrpinternational.fr/quest-ce-quun-backlog-definition-etapes-carac>

15 : Kiwili. <https://www.kiwili.com/fr/Blog/post/la-methode-agile-scrum-les-sprints-ameli>

16 : Scrum, agilité Rockn roll. <https://claudeaubry.fr/release/>.

17 : Hernandez, Rafael. The Model View Controller Pattern MVC Architecture and Frameworks Explained. <https://www.freecodecamp.org/news/themodel-view-controller-pattern-m>

18 : Carton, François. "Interfaces Graphiques : Modèle-Vue-Contrôleur (MVC)." IRIF, Université Paris-Diderot. Disponible sur : <https://www.irif.fr/~carton/Enseignement/InterfacesGraphiques/MasterInfo/Cours/Swing/mvc.html>.

19 : iutenligne.net <https://prive.iutenligne.net/iuRxMOCThIXDIjzG/informatique/langages/kettaf/UML>.

20 : Luccidchart. <https://www.lucidchart.com/pages/fr/langage-uml>.

21 : Eskimoz. "Charte Graphique." Disponible sur : <https://www.eskimos.fr/charte-graphique/>

22 : Définition marketing. <https://www.definitions-marketing.com/definition/logo-ou-logotyp>

Résumé

‘Petracker’ est une application web innovante conçue pour faciliter l’adoption et le sauvetage des animaux domestiques. En réponse aux nombreux défis locaux, tels que l’absence d’une plateforme centralisée, la gestion désorganisée des soins vétérinaires, le nombre croissant d’animaux abandonnés ainsi qu’à la complexité du processus d’adoption ; Petracker offre des solutions efficaces et accessibles. L’application met en avant des profils détaillés des animaux à adopter, facilite la prise de rendez-vous en ligne pour simplifier les visites et les consultations avec des vétérinaires, et intègre un système d’alerte pour les situations d’urgence et les événements de sauvetage. En partageant du contenu communautaire pertinent, Petracker vise à transformer et améliorer le processus d’adoption et le sauvetage des animaux domestiques, tout en favorisant une cohabitation harmonieuse entre les humains et les animaux en Algérie.

Mots clés : Application web, Petracker, Galoufa, HTML, UML, MVC, Scrum, Release, Sprint.

Abstract

‘Petracker’ is an innovative web application designed to facilitate the adoption and rescue of domestic animals. In response to numerous local challenges, such as the lack of a centralized platform, disorganized veterinary care management, the growing number of abandoned animals, and the complexity of the adoption process, Petracker offers effective and accessible solutions. The application features detailed profiles of animals available for adoption, facilitates online appointment scheduling to simplify visits and consultations with veterinarians, and integrates an alert system for emergencies and rescue events. By sharing relevant community content, Petracker aims to transform and improve the adoption and rescue process for domestic animals, while promoting a harmonious coexistence between humans and animals in Algeria.

Keywords : Web Application, Petracker, Galoufa, HTML, UML, MVC, Scrum, Release, Sprint.