

# Introduction générale

Les réseaux sociaux présentent aujourd’hui un moyen fort et très répandu de communication vue les fonctionnalités qu’ils offrent et surtout les interactions en temps réel entre les utilisateurs ce qui leurs permet d’échanger des messages , partager des publications se notifier et de personnaliser leurs profils etc . . .

Le but étant de créer un environnement sociale avec les fonctionnalités nécessaires pour répondre aux divers besoins d’un secteur particulier .

Dans ce cadre, l’entreprise Ng-enious, qui se spécialise dans le développement web et mobile nous a accueilli durant un stage de 4 mois afin de concevoir et développer un réseau social qui a la spécificité d’offrir des usages sociaux au amateurs d’animaux.

Notre projet se définit en tant qu’un réseau sociale dédié aux bénévoles et activistes animaliers.

En effet , notre solution vise à réunir tous les bénévoles , activiste et spécialiste animalier sous une même plateforme. Une plateforme dont le but est de créer un espace qui offre les fonctionnalités nécessaire pour pouvoir intervenir plus rapidement à un animal , échanger des idées et des connaissances d’élevage , partager des conseils , échanger des messages privés et bien plus d’autres fonctionnalités .

Ce rapport de projet de fin d’étude se divise en quatre chapitres.

**Le premier chapitre** s’intitule «Contexte général et étude de l’existant», dans lequel nous donnons une présentation sur l’étude de l’existant et les orientations de travaux. Par la suite, on va faire une étude sur les méthodologies les plus utilisés pour le développement des applications et on va choisir parmi celles la méthode adéquate à notre projet. Et nous terminons par un diagramme de gantt pour planifier les tâches de notre travail.

**Le deuxième chapitre** s’intitule «Capture et spécification des besoins », dans lequel nous étudions les besoins fonctionnels, non fonctionnels et techniques auxquels notre application doit répondre. Par la suite, on va présenter la spécification des besoins fonctionnels à l’aide du diagramme de

cas d'utilisation. Nous achevons ce chapitre par une spécification préliminaires des maquettes.

**Le troisième chapitre** s'intitule «Conception» , dans lequel nous passons à la conception de notre application en se basant sur la méthodologie UML.

**Le quatrième chapitre** s'intitule « Réalisation » dans lequel nous mettons l'accent sur l'environnement de travail, les outils utilisés pour le développement et nous intéressons aussi à l'implémentation informatique de notre application web. Ce chapitre sera considéré comme une validation de toute la démarche adoptée pour réaliser notre projet selon les besoins requis.

Nous terminons ce manuscrit par une conclusion générale, dans laquelle nous présentons , d'abord, un résumé des différentes étapes de la réalisation, et ensuite, nous proposons nos perspectives.

# Table des matières

<b>Introduction générale</b>	<b>i</b>
<b>1 Contexte général et étude de l'existant</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Description société d'accueil . . . . .	1
1.3 Contexte général et problématique . . . . .	2
1.4 Étude de l'existant . . . . .	3
1.4.1 Applications connexes . . . . .	3
1.4.2 Les solutions des réseaux sociaux . . . . .	6
1.4.3 Discussion . . . . .	7
1.5 Solution proposée . . . . .	9
1.6 Méthodologie de réalisation . . . . .	10
1.6.1 Modèle en V . . . . .	10
1.6.2 Modèle en Scrum . . . . .	11
1.6.3 Modèle RUP . . . . .	12
1.6.4 Modèle en 2TUP . . . . .	13
1.6.5 Mise en pratique du processus 2TUP . . . . .	13
1.7 Diagramme de Gantt . . . . .	15
1.8 Conclusion . . . . .	16
<b>2 Capture et spécification des besoins</b>	<b>17</b>
2.1 Introduction . . . . .	17
2.2 Capture des besoins . . . . .	17
2.2.1 Capture des besoins fonctionnels . . . . .	17
2.2.2 Capture des besoins non-fonctionnels . . . . .	19
2.2.3 Capture des besoins techniques . . . . .	20
2.3 Technologies utilisées . . . . .	21
2.4 Spécification des besoins fonctionnels . . . . .	21
2.4.1 UML . . . . .	21
2.4.2 Diagramme de cas d'utilisation . . . . .	22
2.5 Spécification préliminaire des maquettes . . . . .	31
2.5.1 Front Office . . . . .	31
2.5.2 Back Office . . . . .	31

2.6	Conclusion . . . . .	32
<b>3</b>	<b>Conception</b>	<b>33</b>
3.1	Introduction . . . . .	33
3.2	Architecture 2-tiers Firebase . . . . .	33
3.3	Architecture MVC(Modèle-vue-contrôleur) . . . . .	34
3.4	Conception détaillé . . . . .	35
3.4.1	Vue de kruchten . . . . .	35
3.4.2	Conception de cas d'utilisation "Gérer profile" . . . . .	38
3.4.3	Conception de cas d'utilisation "Poster annonce" . . . . .	41
3.4.4	Conception de cas d'utilisation "Gérer fil d'accueil" . . . . .	45
3.4.5	Conception de cas d'utilisation "Contrôler les annonces"	48
3.4.6	Diagramme de classes globale . . . . .	51
3.5	Modèle MVC proposé . . . . .	52
3.6	Conclusion . . . . .	53
<b>4</b>	<b>Réalisation</b>	<b>54</b>
4.1	Introduction . . . . .	54
4.2	Environnement de développement . . . . .	54
4.2.1	NodeJS . . . . .	54
4.2.2	Ionic 3 Framework . . . . .	55
4.2.3	Angular 5 . . . . .	55
4.2.4	Firebase . . . . .	56
4.2.5	Draw.io . . . . .	59
4.2.6	Github . . . . .	59
4.2.7	Inkscape . . . . .	59
4.2.8	Visual Studio Code . . . . .	60
4.3	Présentation de la solution "Hanimo" . . . . .	61
4.3.1	Front Office . . . . .	61
4.3.2	Back Office . . . . .	72
4.4	Conclusion . . . . .	74
<b>Conclusion générale</b>		<b>75</b>

# Table des figures

1.1	Logo de la société ng-enious . . . . .	2
1.2	ASPCA – Emergency Pet Safety . . . . .	3
1.3	Dogalize . . . . .	4
1.4	Pet First Aid - Red Cross . . . . .	5
1.5	Yummy Pets . . . . .	5
1.6	Pages Facebook . . . . .	6
1.7	Groupes Facebook . . . . .	7
1.8	Comparaison technique entre les solutions existantes . . . . .	8
1.9	Comparaison fonctionnelles entre les solutions existantes . . . . .	9
1.10	Description des principaux fonctionnalités . . . . .	10
1.11	Description du processus 2tup . . . . .	14
1.12	Description du planning de travail Diagramme de Gantt . . . . .	16
2.1	Diagramme de cas d'utilisation global . . . . .	23
2.2	Diagramme de cas d'utilisation « Gérer profil » . . . . .	24
2.3	Diagramme de cas d'utilisation « Gérer les contacts » . . . . .	25
2.4	Diagramme de cas d'utilisation « Poster des annonces » . . . . .	26
2.5	Diagramme de cas d'utilisation « Gérer fil d'accueil » . . . . .	27
2.6	Diagramme de cas d'utilisation « Contrôler les annonces » . . . . .	29
2.7	Diagramme de cas d'utilisation «Poster sujet de discussion » . . . . .	30
2.8	Maquette de l'application pour la Front Office . . . . .	31
2.9	Gestion de profil d'un administrateur . . . . .	31
2.10	Visualisation des listes des sujets . . . . .	32
3.1	L'architecture 2-tiers Firebase . . . . .	34
3.2	Model-View-Controller (MVC) . . . . .	34
3.3	Vue de Kruchten . . . . .	35
3.4	Diagramme de séquence «Gérer profile» . . . . .	39
3.5	Diagramme de classes «Gérer profile» . . . . .	40
3.6	Diagramme de composant «Gérer profile» . . . . .	40
3.7	Diagramme de déploiement «Gérer profile» . . . . .	41
3.8	Diagramme de séquence «Poster annonce» . . . . .	42
3.9	Diagramme de classes «Poster annonce» . . . . .	43
3.10	Diagramme de composant «Poster annonce» . . . . .	43

3.11	Diagramme de déploiement «Poster annonce» . . . . .	44
3.12	Diagramme de séquence «Gérer fil d'accueil» . . . . .	46
3.13	Diagramme de classes «Gérer fil d'accueil» . . . . .	47
3.14	Diagramme de composant «Gérer fil d'accueil» . . . . .	47
3.15	Diagramme de déploiement «Gérer fil d'accueil» . . . . .	48
3.16	Diagramme de séquence «Contrôler les annonces» . . . . .	49
3.17	Diagramme de classes «Contrôler les annonces» . . . . .	49
3.18	Diagramme de composant «Contrôler les annonces» . . . . .	50
3.19	Diagramme de déploiement «Contrôler les annonces» . . . . .	50
3.20	Diagramme de classes globale . . . . .	51
3.21	Modèle MVC proposé . . . . .	52
4.1	Logo NodeJs . . . . .	55
4.2	Logo Ionic . . . . .	55
4.3	Logo Angular . . . . .	56
4.4	Logo Firebase . . . . .	56
4.5	Firebase authentication . . . . .	57
4.6	Firebase Realtime Database . . . . .	58
4.7	Firebase Cloud Storage . . . . .	58
4.8	Logo Draw.io . . . . .	59
4.9	Page d'accueil de Github . . . . .	59
4.10	Page d'accueil de Inkscape . . . . .	60
4.11	Logo Visual Studio Code . . . . .	60
4.12	Interface de connexion . . . . .	61
4.13	Interface d'inscription . . . . .	62
4.14	Visualisation de profil de l'utilisateur connecté . . . . .	63
4.15	Interface des paramètres . . . . .	64
4.16	Visualisation de profil d'un ami(e) . . . . .	65
4.17	Envoyer une demande d'ajout en amis . . . . .	66
4.18	Annulation de l'envoi d'une demande d'ajout en amis . . . . .	67
4.19	Profil d'un utilisateur non ami . . . . .	68
4.20	Composer un message . . . . .	69
4.21	Interface de fil d'accueil . . . . .	70
4.22	Choix de publication de divers types d'annonces . . . . .	71
4.23	Interface de connexion . . . . .	72
4.24	Interface de gestion de profil . . . . .	72
4.25	Interface pour ajouter un nouveau sujet . . . . .	73
4.26	Interface des listes des sujets . . . . .	73
4.27	Interface des feedbacks . . . . .	74

# Liste des tableaux

3.1	Description textuelle du cas d'utilisation « Gérer profile » . . . . .	38
3.2	Description textuelle du cas d'utilisation « poster annonce » . . . . .	41
3.3	Description textuelle du cas d'utilisation « Gérer fil d'accueil » . . . . .	45
3.4	Description textuelle du cas d'utilisation « Contrôler les annonces » . . . . .	48

# **Chapitre 1**

## **Contexte général et étude de l'existant**

### **1.1 Introduction**

Dans ce chapitre, nous présentons une étude préalable qui permet de situer le projet dans son contexte et de planifier ses aspects. Cette tâche est très importante avant la réalisation de tout projet. En effet, mener une étude préalable permet de déceler les lacunes du système actuel et d'y remédier. Un projet sans valeur additionnelle n'a pas lieu d'être. D'abord, nous allons présenter le contexte général du projet. Ensuite, nous allons présenter l'entreprise d'accueil et l'équipe qui se charge de ce projet. Ceci nous permettra de présenter l'étude de l'existant et les objectifs dégagés de cette dernière. Enfin, pour conclure cette partie, nous présenterons la méthodologie de travail suivie par l'équipe et le chemin suivi pour aboutir à la finalisation du projet.

### **1.2 Description société d'accueil**

“ng-enious” est une société totalement exportatrice localisée à Monastir. Elle se spécialise dans le domaine de création de sites Web et d'applications mobiles. La société cible une clientèle étrangère et travaille beaucoup au niveau de la sous-traitance de projets. Fondée en 2016, “ng-enious” s'impose dans le marché en proposant des solutions innovantes et sur-mesure pour ses clients. Les locaux de la société sont au Cyber Parc régional de Monastir. Cet espace est comparable à une ruche où le gouvernement offre des locaux de travail pour les nouvelles sociétés spécialisées dans les Technologies de l'Information et de Communication. Étant un espace gouvernemental, le prix de la location des locaux est très réduit permettant ainsi un plus rapide développement pour les nouvelles entreprises.

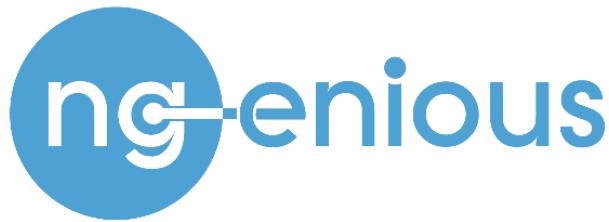


FIGURE 1.1 – Logo de la société ng-enious

L'équipe chargée du projet établi dans ce rapport est formée de deux personnes. Durant ce projet, nous avons été encadrés par Mr Chawki Messaoudi qui a pris le temps de nous guider dans le projet. Toute la collaboration de ce projet fut achevée par l'effort de l'équipe “ng-enious”.

**Adresse :** Bureau 17, Cyber Parc Régional, Rue Ibn Al Jazzar, Skanes - Monastir.

**Site Web :** <http://ng-enious.com>

**Numéro de Téléphone :** (+216) 21 411 187

### 1.3 Contexte général et problématique

Vue l'expansion des communautés et des groupes surtout sur les réseaux sociaux qui s'intéressent au secourisme et intervention au animaux ( surtout de compagnie ) en situation de danger tel que perdus , errant ou même nécessitant une intervention de spécialiste , la fonction de secourisme et intervention au animaux est devenu une fonction importante nécessitant une organisation sans faille permettant la communication , le partage de service et l'intervention ainsi la déclaration des cas d'urgence pour obtenir de l'aide et sauver la vie de ces créatures.

Le projet vise la création d'une communauté de bénévolat pour les animaux bien informée et collaborative. L'idée est de proposer une solution en offrant des possibilités de réseautage accessibles aux bénévoles et activistes animaliers. En effet, actuellement pour aider un animal , on doit avoir un bon réseau de connaissances des vétérinaires et activistes via les réseaux sociaux ou la vie quotidienne. Sinon, il faut rejoindre les groupes et les pages de bénévolat pour les animaux pour être au courant. La solution que nous proposons est une plateforme de réseau sociale qui permet à ces activistes animalier de collaborer et échanger pour venir en aide à des animaux en cas d'urgence . Nous détaillerons la solution dans la suite .

## 1.4 Étude de l'existant

Dans cette partie nous allons citer quelques solutions proches de notre projet ou on va dégager les fonctionnalités offertes par chaque solution ce qui nous permettra d'établir une comparaison entre eux . Les solutions qu'on va considérer dans cette partie sont les applications proches et en plus les réseaux sociaux car ces derniers offrent des fonctionnalités intéressantes et importantes dans ce domaine .

### 1.4.1 Applications connexes

D'après nos recherches, surtout sur le marché tunisien, ils n'existent pas de vraies solutions répondant au besoin qui est la mise en relation et l'organisation des communautés et individus pour permettre l'intervention et le secourisme des animaux . Cependant à sur échelle mondiale, il existe quelques applications .

#### ASPCA – Emergency Pet Safety

ASPCA (American Society for the Prevention of Cruelty to Animals) application pour la prévention de la cruauté envers les animaux vous permet de stocker les dossiers médicaux de votre animal de compagnie. Il vous indique également comment prendre soin des animaux de compagnie lors des catastrophes et vous permet d'accéder à des modèles d'affiches numériques «Missing Pet».

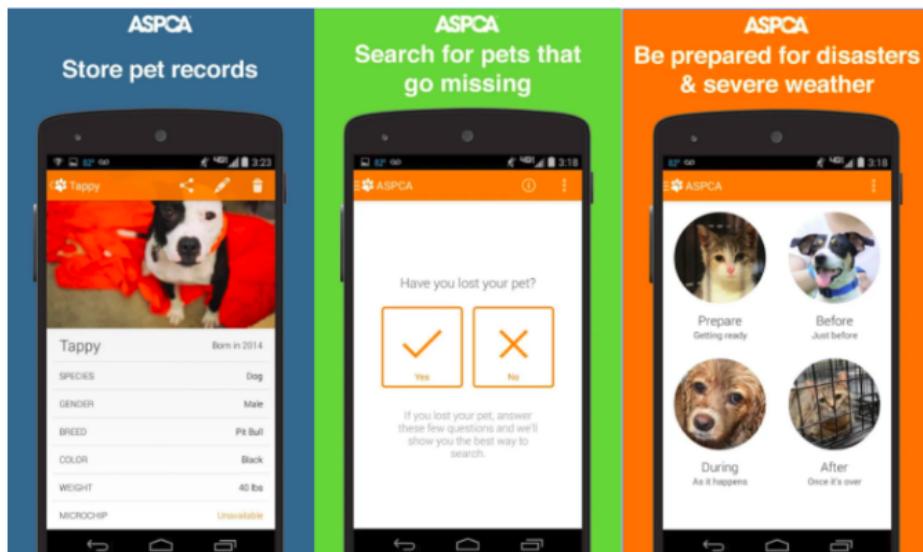


FIGURE 1.2 – ASPCA – Emergency Pet Safety

### Fonctionnalités offertes

- Déclarer la perte d'animal de compagnie.
- conseils pour prendre soin de l'animal.
- stocker un dossier médical de l'animal.

### Dogalize

Dogalize est une application destinée aux activistes animalier qui offre le partage de publication ou de statuts. Vous pourrez également obtenir de l'aide auprès des vétérinaires, discuter avec vos amis, demander des conseils à la communauté. Avec les cartes Dogalize, ou se trouve des activités et des endroits accueillants pour les animaux de compagnie

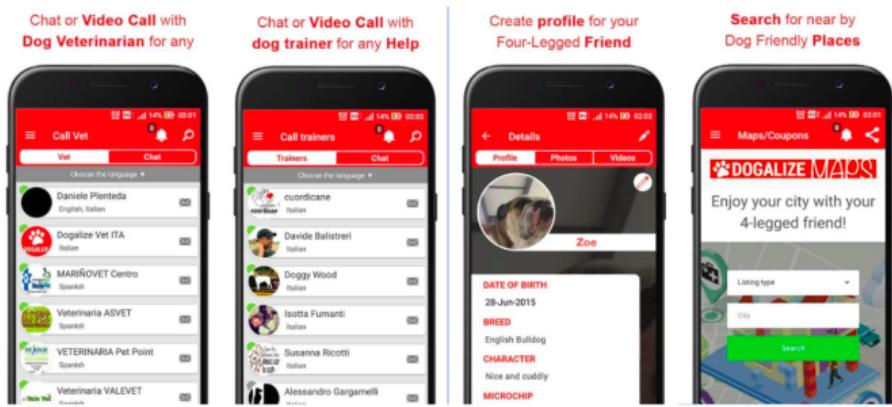


FIGURE 1.3 – Dogalize

### Fonctionnalités offertes

- Partage de publication ou de statuts
- Trouver des conseils.
- Échanger messages avec les autres utilisateurs.
- Obtenir l'aide des spécialistes tel que vétérinaire, dresseur, ...
- Trouver des foyers d'animaux proches ou même des espaces d'activité.

### Pet First Aid - Red Cross

L'application de secourisme pour animaux de compagnie de la Croix-Rouge américaine met des conseils vétérinaires pour les urgences quotidiennes dans la paume de main. Avec des vidéos, des quiz interactifs et des conseils simples.

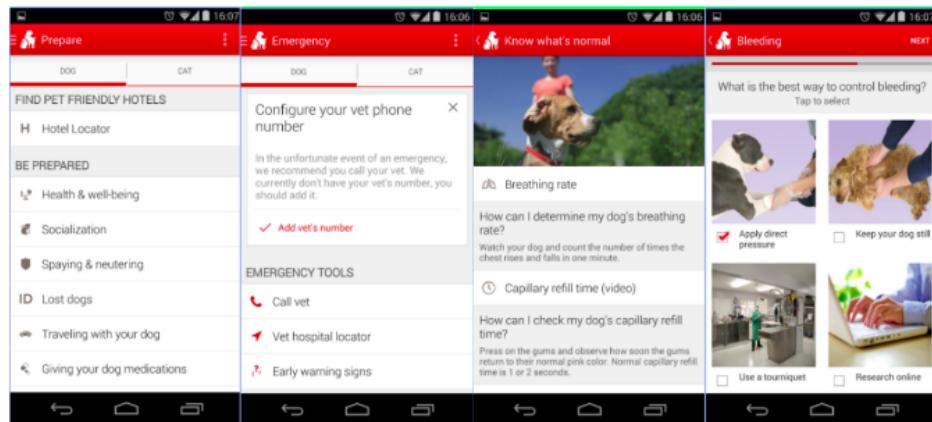


FIGURE 1.4 – Pet First Aid - Red Cross

#### Fonctionnalités offertes

- Obtenir l'aide des vétérinaires
- Consulter des conseils d'animalerie
- Localisez votre hôpital vétérinaire d'urgence ou animaux de compagnie hôtels les plus proches
- Personnaliser les profils d'animaux multiples et nominations vétérinaires établies.

#### Yummypets

C'est une plateforme communautaire au service du propriétaire de l'animal de compagnie. Elle offre plusieurs service d'animalerie



FIGURE 1.5 – Yummy Pets

#### Fonctionnalités offertes

- Permettre aux utilisateurs d'échanger des nouveautés
- Conseils pour prendre soin des animaux de compagnie
- Vente de produits d'animalerie

#### 1.4.2 Les solutions des réseaux sociaux

Un réseau social est une structure formée par des relations entre des personnes. C'est un regroupement d'individus ou d'organisations qui discutent et interagissent entre eux. Ils partagent des opinions, des idées ou encore du contenu multimédia. Le plus connus et le plus utilisé des réseaux sociaux à présent est Facebook .

##### Facebook



FIGURE 1.6 – Pages Facebook



FIGURE 1.7 – Groupes Facebook

Tant que Facebook permet aux utilisateurs de créer des pages , des groupes, des événements gratuitement, les activistes animalier ont appuyé sur ces fonctionnalités pour créer des communautés a travers Facebook , en créant des pages, des groupes et en lançant des événements pour ceux qui s'intéressent aux bénévoles pour les animaux ainsi ils ont pu partager leurs intérêt envers les animaux et déclarer des états d'urgences pour obtenir de l'aide.

Pourtant cette solution reste manquante et incomplète pour couvrir les besoins des activistes animalier puisque Facebook est un réseau social grand public et n'est pas destiné pour ce genre d'activités.

#### 1.4.3 Discussion

Les solutions existantes de plateforme d'intervention et de secourisme regroupent plusieurs fonctionnalités et chacune offre des fonctionnalités intéressantes mais un peu limitées et parfois payantes. Mais une telle plateforme n'existe pas encore qui englobe la plupart des fonctionnalités et y ajoutes d'autres. Dans ce qui suit nous allons plus détailler les différences technique et fonctionnelle ainsi tirer les avantages et inconvénients de chaque solution.

Nous représentent ci-dessous une étude comparatif des critères techniques et fonctionnelles des applications connexes et solutions des réseaux sociaux

## Comparaison technique

Pour effectuer une critique technique nous avons appuyer notre comparaison sur des critères qu'on trouve essentielle pour toutes application . Ci dessous , nous définissons chaque critère .

### Les critères Techniques :

**Ergonomie et graphisme :** Le même graphisme sur tous les OS.

**Mise a jour du contenu :** Simple et immédiate.

**Simplicité :** L'utilisation assez simple indépendamment du connaissance de l'utilisateur en informatique.

**Exhaustivité :** L'application doit satisfaire les besoins des utilisateurs (La recherche, la demande, validation de qualité de service...).

**Qualité de service :** Elle doit rendre ce service efficacement (de manière complète, pertinente, être toujours disponible, fiable, être pérenne et peu cher, facile et agréable à prendre en main et à utiliser...).

**Accès conditionnées :** L'application doit garantir la confidentialité et la protection des données personnelles.

**Rapidité :** Vitesse de traitement et de réponse aux évènements

**Performance :** Capacité à réaliser et répondre rapidement aux demandes des utilisateurs.

**Evolutivité/Agilité :** Capacité à évoluer vite et sur spécifications chanches.

	ASPCA Emergency	Dogalize	Pet first aid	Yummy Pets	Facebook
Compatibilité multiplateforme	(+/-)	(+/-)	(+/-)	(+/-)	(+)
Ergonomie et graphisme	(+)	(+)	(-)	(-)	(+)
Mise a jour du contenu	(+)	(+)	(+)	(+)	(+)
Simplicité	(+)	(+)	(+)	(+)	(+)
Exhaustivité	(+)	(+)	(+/-)	(+)	(+)
Qualité de service	(+)	(+)	(+)	(+)	(+)
Accès conditionnées	(+/-)	(+)	(-)	(+)	(+)
Rapidité	(+/-)	(+)	(+/-)	(+)	(+)
Performance	(+)	(+)	(+/-)	(+)	(+)
Evolutivité/Agilité	(+)	(+)	(+/-)	(+)	(+)

FIGURE 1.8 – Comparaison technique entre les solutions existantes

## Comparaison technique

### Critères fonctionnelle :

	ASPCA Emergency	Dogalize	Pet First Aid - Red Cross	Yummy Pets	Facebook
Authentification	✗	✓	✓	✓	✓
Gestion de profil	✗	✓	✓	✓	✓
Poster des annonces	✗	✗	✗	✗	✗
Signaler une urgence	✗	✓	✗	✗	✗
Mise à jour de fil d'actualité	✗	✓	✗	✓	✓
Création de contenu privé	✗	✗	✗	✗	✓
Messagerie instantanée	✗	✓	✗	✗	✓
Recherches filtrées	✓	✓	✓	✓	✓
Participer aux sujets de discussion ou événement	✗	✗	✗	✓	✗
Systèmes réputation de profile	✗	✗	✗	✗	✗

FIGURE 1.9 – Comparaison fonctionnelles entre les solutions existantes

## 1.5 Solution proposée

Suite à la définition des réseaux sociaux et d'après les avantages qu'offre un réseau social, notre solution consiste en une plateforme sous forme d'un réseau sociale permettant de venir en aide à des animaux perdus, errants ou à donner. Cette plateforme est dédié aux personnes qui défendent les animaux et leur droits appelés aussi Pet Lovers ainsi que les communautés et groupes qui apportent le plus dans le domaine de secourisme d'animaux.

Notre solution offre une partie public front-office accessible par tous utilisateurs et une partie privé à l'administration back-office accessible pour les administrateurs qui permet le contrôle .

Dans la figure ci dessous nous essayerons de résumer les principaux fonctionnalités de notre solution .

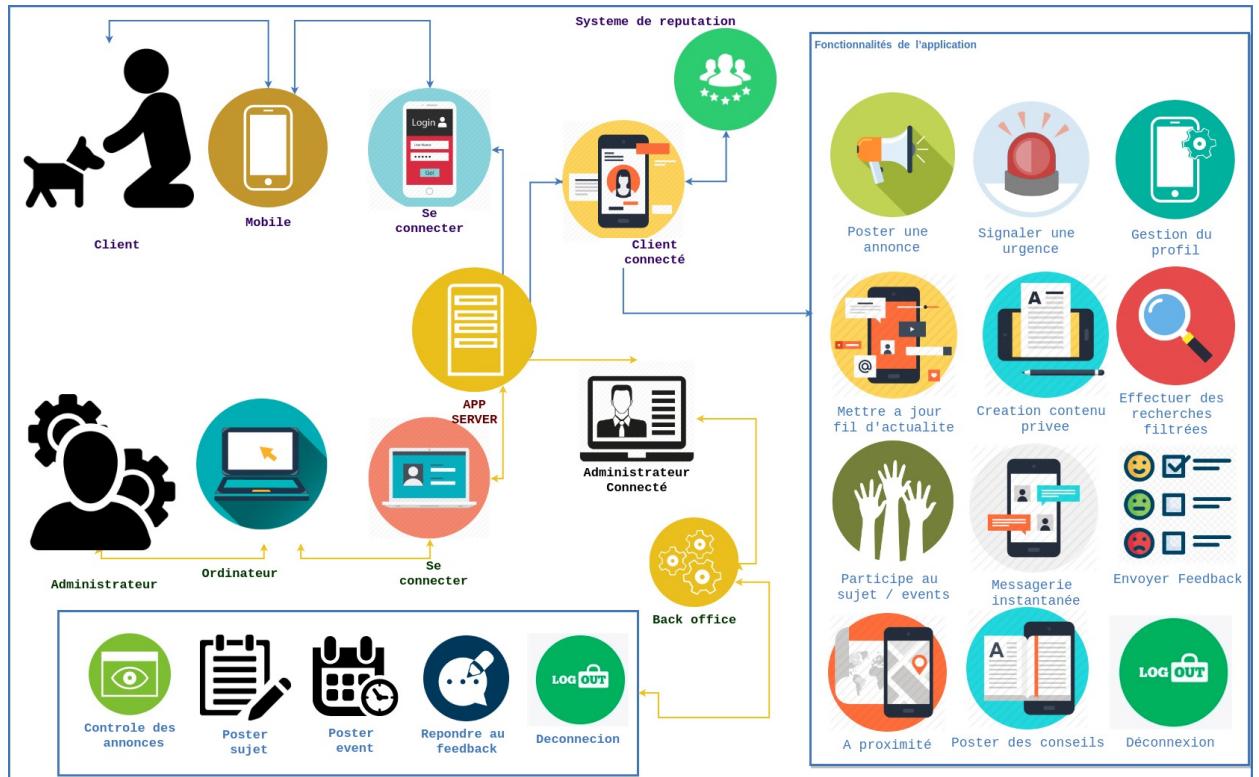


FIGURE 1.10 – Description des principaux fonctionnalités

## 1.6 Méthodologie de réalisation

Une méthodologie de développement est un cadre utilisé pour structurer, planifier et contrôler le développement d'une application. Suivre une méthodologie de développement permet de baser un projet sur un modèle de conception et des formalismes de notations ce qui implique la réduction des coûts de développement et encourager la communication entre les personnes . Il existe une diversité de méthodologies disponibles et chacune offre des avantages donc choisir une n'est pas une tâche facile . Pour ça nous procéderons à une étude à propos des méthodes existantes pour dégager la méthodologie la plus adaptée à notre équipe et à notre projet .

### 1.6.1 Modèle en V

Le cycle en V est un paradigme du développement informatique. Il décrit les étapes essentielles du développement d'un logiciel, le cycle de vie du projet. Il est représenté par un V dont la branche descendante contient toutes les étapes de la conception du projet, et la branche montante toutes les étapes de tests du projet. La pointe du V, quant à elle, représente la

réalisation concrète du projet, le codage ; on pourrait donc en déduire, de manière simpliste, que les deux branches montantes et descendantes ne sont que de la documentation.

**Avantages :**

- Meilleure validation intermédiaire(Bon suivi de projet, favorise la décomposition fonctionnelle de l'activité).
- Modèle éprouvé très utilisé pour de grands projets.
- Facile de prévoir les tests à réaliser au moment où l'on conçoit une fonctionnalité ou une interface.
- Permet d'espérer que le livrable final sera parfait, puisque les étapes de test sont aussi nombreuses que les étapes de réflexion.
- Il est facile de prévoir les tests à réaliser au moment où l'on conçoit une fonctionnalité ou une interface.
- Le travail s'enchaîne de façon assez naturelle.

**Inconvénients :**

- Rarement utilisé tel quel et le V est bien souvent déséquilibré .
- Un modèle toujours séquentiel.
- Adapté aux problèmes bien connus c'est à dire il est idéal quand les besoins sont bien connus, quand l'analyse et la conception sont claires.
- Le manque de souplesse : chaque phase doit être terminée (spécification, conception, développement) avant de passer à la suivante. Chacune des étapes va durer plus longtemps et donc coûter plus chère.
- La péremption du produit : sur de gros projets dont la durée de réalisation est de plusieurs années.

### 1.6.2 Modèle en Scrum

Scrum est aujourd'hui la méthode agile populaire. Elle se caractérise par itérations assez courts et un formalisme réduit. Le cycle de vie dans un projet SCRUM est composé de : Phase d'initialisation :Phase linéaire aux processus explicitement connus et dont les inputs et les outputs sont bien déterminés. Phase de cloture : est aussi une phase linéaire dont le processus sont clairement définis.

**Avantages :**

- Personnel engagé : l'une des caractéristiques de SCRUM, c'est que le personnel participe activement à la définition des activités et des horaires, de sorte que le degré d'engagement et la motivation sont plus élevé.
- Simplicité des processus.
- Augmentation de productivité.
- Amélioration de la communication.
- Flexibilité de changement.
- Simplicité, efficacité et qualité.
- Esprit d'équipe.

#### **Inconvénients :**

- Violation de responsabilité.
- Cette méthodologie a besoin de membres expérimentés de l'équipe seulement. si l'équipe se compose de personnes qui sont novices, le projet ne peut pas être achevé à temps.
- Faible documentation et donc facile à détourner.

#### **1.6.3 Modèle RUP**

Rational Unified Process, instanciation par Rational Software (IBM) des préceptes UP(processus unifié). Le principe de cette méthode est de parcourir un cycle de vie durant une itération. Chaque phase du cycle de vie est très précisément détaillé. C'est une méthode générique, itérative et incrémentale. Il divise le processus de développement en quatre phases : Inception : L'idée de ce projet est déclarée. L'équipe de développement détermine si le projet mérite d'être poursuivi et quelles ressources seront nécessaires. Elaboration : L'architecture du projet et les ressources nécessaires sont encore évaluées . Les développeurs considèrent les applications possibles du logiciel et les associés au développement. Construction : Le projet est développé est complété. Le logiciel est conçu, écrit et testé. Transition : Le logiciel est diffusé au public. Les derniers ajustements ou mise à jour sont effectués en fonction des commentaires finaux des utilisateurs.

#### **Avantages :**

- RUP est l'une des plus célèbres implémentations de la méthode PU permettant de donner un cadre au développement logiciel.
- Le RUP est plus documenté et donc peut être plus guidé.
- Traçabilité à partir des Use Cases jusqu'au déploiement.
- Approche basé sur l'architecture.
- Gestion des risques dans les projets.
- Cadre propice à la réutilisation
- Moins couteux de changer la portée
- Certaines fonctionnalités de travail peuvent être développées rapidement et tout dans le cycle de vie.

#### **Inconvénients :**

- Coût de personnalisation souvent élevé.
- Plus de ressources peuvent être nécessaires.
- Chaque phase d'une itération est rigide sans chevauchements.
- Plus de ressources peuvent être nécessaires.
- Vision non évidente.
- Processus très axé.

#### 1.6.4 Modèle en 2TUP

La méthode 2TUP préconise un cycle de vie en Y qui dissocie et parallélisé la résolution des questions fonctionnelles et techniques. Le cycle de vie de 2TUP s'apparente à un cycle en cascade mais introduit une forme itérative interne à certaines tâches.

##### Avantages :

- Modèle d'analyse réutilisable.
- Itératif.
- Indépendant de la taille du projet.
- Prise en compte de la gestion et du technologique.
- Ne définit pas les typologies d'intervenants et les livrables.

##### Inconvénients :

- Il y a un manque d'accent sur la conception et la documentation nécessaires.
- Couvre seulement la partie développement.
- Pas de documents types.

**Synthèse de méthodologie** Après cette comparaison, nous constatons que la méthodologie 2TUP est la plus efficace pour réaliser les différentes étapes de développement de notre application.

#### 1.6.5 Mise en pratique du processus 2TUP

Le 2TUP propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels. Le cycle de vie de 2TUP s'apparente à un cycle en cascade, mais introduit une forme itérative interne à certaines tâches . Nous avons choisi ce processus vue son approche nouvelle, originale et surtout son adaptation aux solutions des réseaux sociaux . Le processus 2TUP se résume clairement dans la figure suivante :

Le processus 2TUP signifie que le processus suit deux chemins . Il s'agit du «chemin fonctionnels » et du «chemin technique », qui correspondent aux deux axes de changement imposés au système d'information.

1. Branche fonctionnelle ou « gauche » Elle vise la capture des besoins fonctionnels et l'analyse des spécifications fonctionnelles de manière à déterminer ce que va réaliser le système en terme de métier. C'est ici, qu'on identifie et dégage toutes les fonctionnalités du système à réaliser.
2. Branche technique ou « droite » Elle permet la capture des besoins non fonctionnels. Il s'agit essentiellement des contraintes que l'application doit prendre en compte comme par exemple les contraintes d'intégration, les contraintes de développement et les contraintes de performances.
3. Branche du milieu ou phase de réalisation A l'issue des évolutions du modèle fonctionnel et de l'architecture technique, la réalisation du

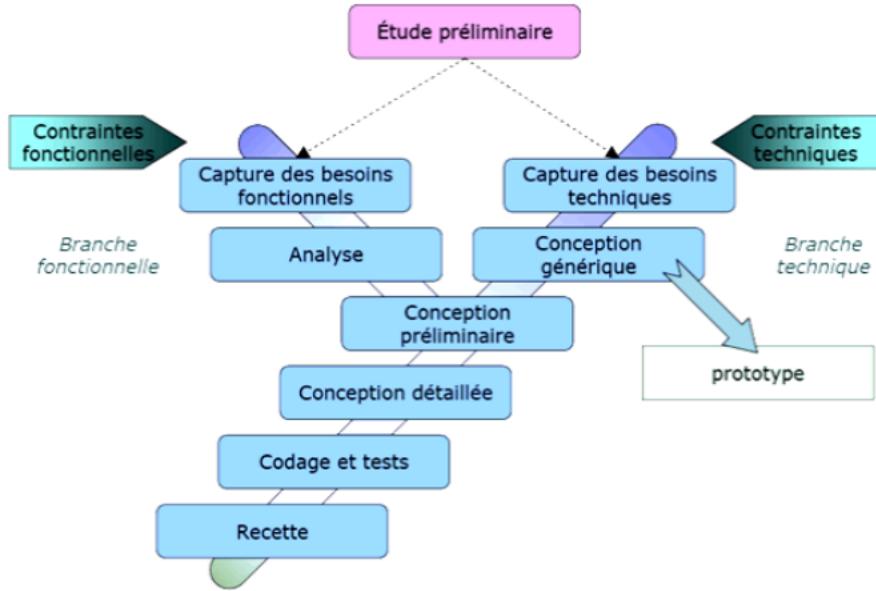


FIGURE 1.11 – Description du processus 2tup

système consiste à fusionner les résultats des 2 branches. Cette fusion conduit à l'obtention d'un processus en forme de Y.

La méthodologie 2TUP se base sur le processus UP qui est de type adaptatif il est de plus :

- **Itératif et incrémental** : Le projet est découpé en des itérations de courte durée. Ces itérations aident à mieux suivre l'avancement du système global. A chaque itération, il est produit un exécutable de façon incrémentale.
- **Piloté par les risques** : Il est identifié et écarté au plutôt tout risque pouvant conduire à un échec du projet. Le pilotage par les risques c'est :
  - Analyser les risques potentiels au plus tôt
  - Hiérarchiser les risques
  - Associer un ensemble des cas d'utilisation à chaque risque
  - Déclencher les itérations selon la criticité des uses cases qu'elles regroupent
  - UP propose une gestion des risques. Ce qui constitue une avancée significative.
- **Piloté par les uses cases** : Pour servir les attentes des utilisateurs, on centre le processus de développement sur leurs besoins. On fait apparaître ces besoins à l'aide de la technique des cas d'utilisation : à l'aide d'une capture des besoins fonctionnels d'un système et en

orientant le travail de chaque itération. Ils vont guider le processus à travers l'utilisation des différents modèles UML qui représentent le système.

- **Centré sur l'architecture** : Le système est décomposé en modules pour des besoins de maintenabilité et d'évolutivité. L'architecture doit prévoir la réalisation de tous les uses case et doit évoluer avec eux. Elle le fait en tenant compte de facteurs tels que :
  - La plateforme d'exécution : matériel, système, BD, réseau, etc.
  - Les composants réutilisables : librairies, caisses à outils, composants du commerce, etc.
  - Les considérations de déploiement et les besoins non-fonctionnels : la performance, la fiabilité, la robustesse, etc.
- **Conduit par les cas d'utilisation** : Le processus met en avant les besoins et exigences des futurs utilisateurs du système.

## 1.7 Diagramme de Gantt

Le diagramme de Gantt, couramment utilisé en gestion de projet, est l'un des outils les plus efficaces pour représenter visuellement l'état d'avancement des différentes activités (tâches) qui constituent un projet. La colonne de gauche du diagramme énumère toutes les tâches à effectuer, tandis que la ligne d'en-tête représente les unités de temps les plus adaptées au projet (jours, semaines, mois etc.).

Chaque tâche est matérialisée par une barre horizontale, dont la position et la longueur représentent la date de début, la durée et la date de fin. Ce diagramme permet donc de visualiser d'un seul coup d'œil toutes les tâches à accomplir pour mener le projet à bien, et indique la date à laquelle ces tâches doivent être effectuées (le planning).

- Les différentes tâches à envisager
- La date de début et la date de fin de chaque tâche
- La durée escomptée de chaque tâche
- La date de début et la date de fin du projet dans son ensemble



FIGURE 1.12 – Description du planning de travail Diagramme de Gantt

## 1.8 Conclusion

Dans ce chapitre nous avons abordé la problématique présente puis on a procéder a l'étude de l'existant qui nous a permit de dégager les défauts technique et fonctionnelle des solutions présentes .

En suite nous avons présenté notre solutions ainsi que les objectives qu'on vise atteindre .

Enfin nous avons étudié les méthodologies de développement possible pour aboutir à la méthode 2TUP qu'on a bien expliquer avec une analyse détaillée.

## Chapitre 2

# Capture et spécification des besoins

### 2.1 Introduction

Dans ce chapitre, nous présentons les objectifs de notre application, ce qui nous amène à identifier les fonctionnalités du système en recensant les besoins fonctionnels, de connaître la liste des contraintes et exigences traduite par les besoins non fonctionnels et les besoins techniques qu'on va utiliser pour répondre aux fonctionnalités de notre projet. La spécification des besoins doit décrire sans ambiguïté l'application à développer et toutes les personnes impliquées dans le projet doivent avoir accès à la spécification des besoins.

### 2.2 Capture des besoins

Dans cette partie nous allons détailler nos besoins qui se divisent en 3 types : besoins fonctionnels, besoins non fonctionnels et enfin les besoins techniques.

#### 2.2.1 Capture des besoins fonctionnels

Notre application offre une partie public appelé front-office accessible par tous utilisateurs et une partie privé à l'administration appelé back-office accessible pour les administrateurs pour permettre les tâches de contrôles et chaque partie offre des fonctionnalités distinctes de l'autre

##### Front-Office

**Authentification :** Permettre aux utilisateurs de créer leurs comptes pour se connecter avec leur adresse mail et mot de passe ou s'authentifier via leur compte des réseaux sociaux ( Facebook , Gmail ).

**Compléter les informations** du profil en ajoutant des informations secondaires.

**Poster des annonces à propos d'un animal selon ses conditions :**

- **Besoin d'aide (A3)** : L'animal a besoin d'aide d'une personne à proximité.
  - La notification de ces annonces atteint les utilisateurs dans une zone géographique limitée à 2 Km .
- **Besoin d'un foyer (A2) :** L'animal a besoin d'un foyer d'accueil .
  - La notification de ces annonces atteint les utilisateurs dans une zone géographique limitée à 4 Km .
- **Besoin d'un spécialiste (A1) :** L'animal est blessé , malade et une intervention d'un spécialiste est nécessaire
  - La notification de ces annonces atteint les utilisateurs dans une zone géographique limitée à 8 Km
- **Animal en danger (A0) :** l'animal est en danger et une intervention immédiate est nécessaire
  - La notification de ces annonces atteint les utilisateurs dans une zone géographique limitée à 10 Km .

**NB :** Pour poster une annonce de type A1 ou A0 l'utilisateur doit disposer d'au moins 30 points de réputation.

**Signaler une urgence :** le signale d'urgence se fait lorsqu'un utilisateurs poste une annonce de type A1 ou A0 . ainsi ce type d'annonce serait visible en premier et la notification atteindra les utilisateurs dans une zone géographique plus large que les autres types d'annonces (A2 , A3).

**Gérer le profil :** chaque utilisateur aura un profil consultable ou il peut ajouter des conseils et des publications. Il peut aussi accéder aux paramètres pour changer ses informations du profil et du compte.

**Consulter fil d'accueil :** Un utilisateur peut visualiser les publications partagées par ses amis, les commenter, les aimer et sauvegarder des conseils.

**Gestion du contenu privé :** Visualise les conseils sauvegardés récemment

**Messagerie instantanée :** l'application permet l'échange de messages instantanés entre les utilisateurs.

**Envoyer des feedbacks :** Les utilisateurs peuvent envoyer des feedbacks aux administrateurs pour suggérer des idées , porter plaintes , donner son avis.

**Participer aux sujets de discussions :** Les administrateurs de l'application peuvent publier des sujets de discussion en relation avec des événements

internationaux a propos des animaux où les utilisateurs pourront participer à ces sujets en commentant leurs avis , des solutions , échanger des idées.

**Systèmes de réputation de profil :** Le système de réputation de profil est un système qui permet à l'utilisateur d'évoluer son profil en augmentant ses points de réputation ce qui lui permet plus de priviléges.

Dans notre application , initialement l'utilisateur commence par une réputation égale à 1 , puis à chaque action effectué il augmente ses point de réputation :

- Completer toute les informations du profil : +5 pts
- Publier des conseils : +3pts
- Poster des annonces
  - Type A3 : +4pts
  - Type A2 : +6pts
  - Type A1 : +10pts
  - Type A0 : +12pts
  - Autre : +1pts
- Commenter / Aimer une annonce + 2pts
  - Propriétaire de l'annonce : +2pts
- Partager une annonce : +4pt
  - Propriétaire de l'annonce : +4pts
- Participation au sujet de discussion :
  - Commenter : +2pts
  - Partager le sujet : +3pts

### Back-Office

**S'authentifier :** L'administrateur peut s'identifier avec son adresse mail et un mot de passe

**Contrôle des annonces :**

- Majorer ou minorer l'état d'urgence d'une annonce .
- Supprimer une annonce

**Gérer des sujet de discussion :** L'administrateur peut poster un sujet de discussion . Ce poste sera visible à tous les utilisateurs . il peut aussi modifier ou supprimer un sujet .

**Répondre au feedback :** L'administrateur peut consulter les messages envoyé par les utilisateurs ( feedback , plainte , avis , ... ) et répondre .

#### 2.2.2 Capture des besoins non-fonctionnels

Les besoins non-fonctionnels représente les points qui agissent sur la satisfaction et le rendement de l'utilisateur de façons indirectes . notre application doit être à la mesure des points suivants :

**Ergonomie et graphisme** : Le même graphisme sur tous les OS.

**Mise à jour du contenu** : Simple et immédiate.

**Simplicité** : L'utilisation assez simple indépendamment de la connaissance de l'utilisateur en informatique.

**Performance** : Capacité à réaliser et répondre rapidement aux demandes des utilisateurs.

**Accès conditionnées** : L'application doit garantir la confidentialité et la protection des données personnelles.

**Rapidité** : Vitesse de traitement et de réponse.

### 2.2.3 Capture des besoins techniques

Technologie les plus utilisé pour le développement des applications mobile :

Technologie	Langage	Avantages	Inconvénients
Android	Java / Kotlin	<ul style="list-style-type: none"><li>— Langage de programmation facile à maîtriser</li><li>— Plateforme de développement ouverte</li><li>— Environnement de développement complet</li></ul>	<ul style="list-style-type: none"><li>— N'est pas multiplateforme</li><li>— Le développement d'une application prend plus du temps par rapport aux autres plateformes</li><li>— Environnement puissant nécessitant l'utilisation d'un matériel puissant</li></ul>
IOS	Objective-C/Swift	<ul style="list-style-type: none"><li>— Environnement de développement complet</li><li>— Sécurité garantie</li></ul>	<ul style="list-style-type: none"><li>— Langage de programmation pas trop répandue comme java</li><li>— Plus coûteux que les autres plateformes</li></ul>
Ionic	Html + Css + TypeScript	<ul style="list-style-type: none"><li>— Langage de programmation facile et maîtrisable , le même du développement web</li><li>— Multiplateforme , un seul code executable sur ios et android</li><li>— Disponibilité des plugins</li><li>— Composant prêt à l'utilisation .</li></ul>	<ul style="list-style-type: none"><li>— Ne supporte pas le développement des applications 3D</li><li>— Nécessite une connaissance du framework Angular</li></ul>

Technologie les plus utilisé pour le développement des applications web :

Technologie	Description	Avantages	Inconvénients
PHP (Back End)	C'est un langage interprété exécuté du côté serveur et non du côté client. La syntaxe de langage provient de celles du langage C, du Perl, du java. La possibilité d'inclure le script PHP au sein d'une page HTML.	<ul style="list-style-type: none"> <li>— Il est gratuit.</li> <li>— Il ne nécessite pas beaucoup de code pour obtenir un résultat.</li> <li>— L'hébergement du PHP est supporté presque partout.</li> <li>— Aucun typage des variables.</li> </ul>	<ul style="list-style-type: none"> <li>— Manque de vision pour le développement de gros projet.</li> <li>— Beaucoup de failles de sécurité</li> </ul>
Angular (Front End)	C'est une framework de développement d'application web coté client (front End) en utilisant le langage typescript en plus du html et css	<ul style="list-style-type: none"> <li>— Amélioration de la réactivité de l'interface de l'utilisateur</li> <li>— Ouvert</li> <li>— Grande communauté et très utilisés</li> <li>— Langage de programmation simple</li> </ul>	-architecture de l'application un peu complexe

## 2.3 Technologies utilisées

Les technologies que nous allons utiliser sont les suivantes :

- Firebase comme un système de back end de gestion et de manipulation de données .
- Le Framework ionic 3 coté front end de l'application mobile .
- Le Framework Angular 5 coté front end du site web du back office .

## 2.4 Spécification des besoins fonctionnels

### 2.4.1 UML

UML (Unified Modeling Language) est une méthode de modélisation orientée objet développée en réponse à l'appel à propositions lancé par

l'OMG (Object Management Group) dans le but de définir la notation standard pour la modélisation des applications construites à l'aide d'objets. C'est une méthode utilisant une représentation graphique. L'usage d'une représentation graphique est un complément excellent à celui de représentations textuelles. En effet, l'une comme l'autre sont ambiguës mais leur utilisation simultanée permet de diminuer les ambiguïtés de chacune d'elle. Il est couramment utilisé en développement logiciel et en conception orienté objet

#### 2.4.2 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation décrit les principales fonctionnalités du système offertes aux acteurs .

Dans notre application nous avons 3 types d'acteurs :

- Invité : cet acteur est invité à créer un compte ou se connecter pour pouvoir accéder à l'ensemble des services de l'application
- Utilisateur : C'est l'acteur principal , il possède un compte et peut exploiter les fonctionnalité du front office cité ci dessus via l'application mobile.
- Administrateur : il possède un compte privé qui lui permet d'exploiter les fonctionnalités du back office via le site web .

#### Diagramme de cas d'utilisation global

La figure ci-dessous représente le diagramme de cas d'utilisation global qui donne une vue générale de comportement fonctionnel de notre application mobile/web. Elle représente les acteurs en interaction avec le système ainsi que ces différentes fonctionnalités.

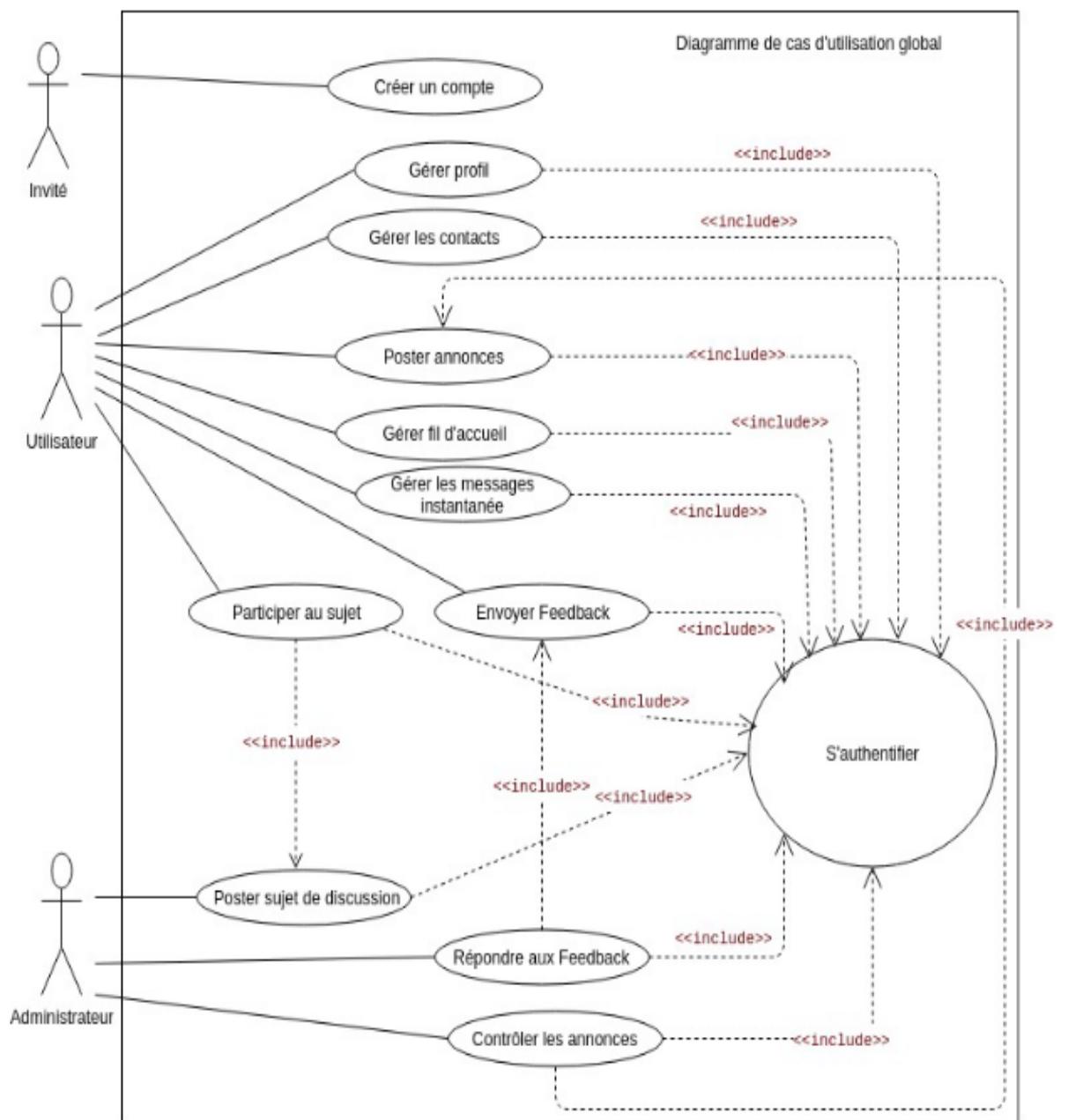


FIGURE 2.1 – Diagramme de cas d'utilisation global

## Cas d'utilisation : Gérer profil

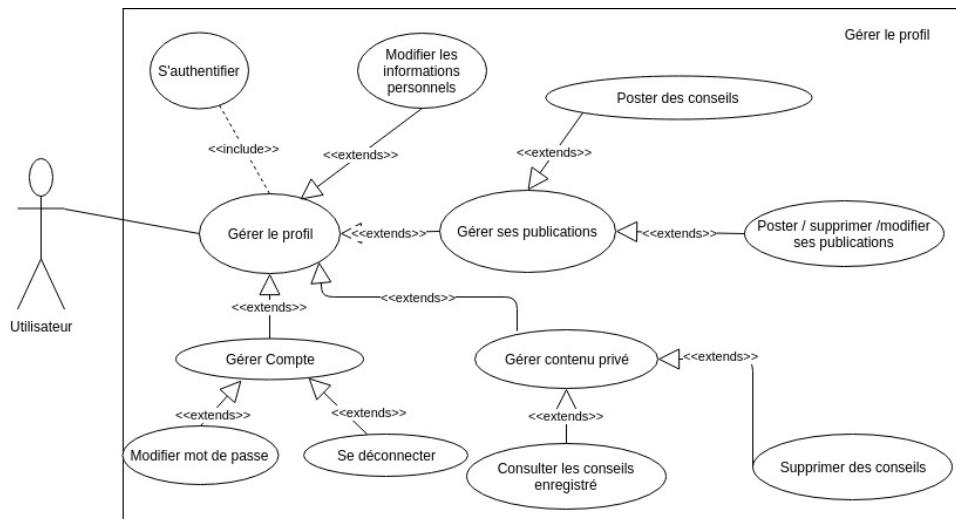


FIGURE 2.2 – Diagramme de cas d'utilisation « Gérer profil »

Titre	Gérer le profil
Description	L'utilisateur a la possibilité de gérer son profil
Acteur	Utilisateur .
Pré-Condition	S'authentifier.
Scénario normal	
1- Modifier ses informations personnelles 2- Gérer ses publication 3- Gérer Compte	<p>L'utilisateur peut modifier ses informations personnel tel que numéro de téléphone , langue , adresse ou description .</p> <ul style="list-style-type: none"> <li>— Créer conseil</li> <li>— Gérer contenu privée</li> <li>— Gérer des publications normales</li> </ul> <p></p> <ul style="list-style-type: none"> <li>— Modifier mot de passe</li> <li>— Se déconnecter</li> </ul>
Post-Condition	Profil géré
Exception	Pas d'accès internet .

## Cas d'utilisation : Gérer les contacts

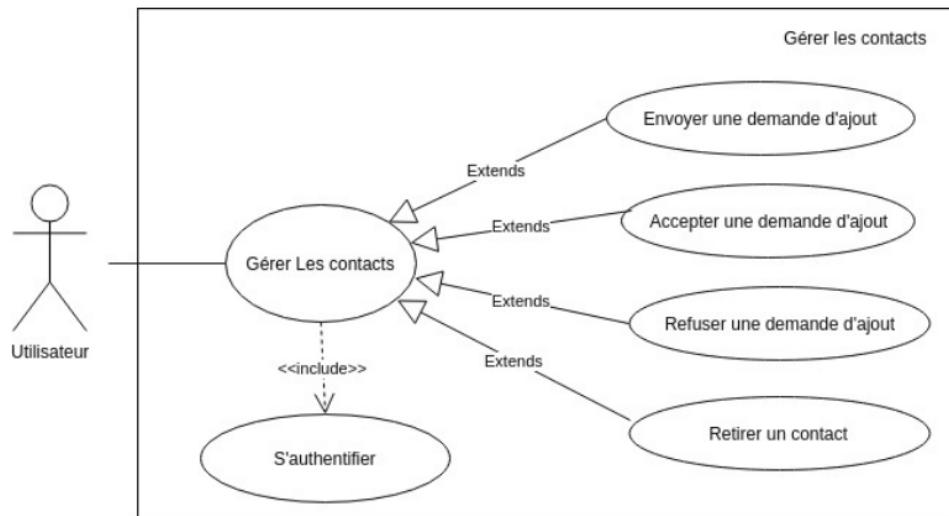


FIGURE 2.3 – Diagramme de cas d'utilisation « Gérer les contacts »

Titre	Gérer les contacts
Description	L'utilisateur a la possibilité de gérer ses contacts
Acteur	Utilisateur
Pré-Condition	S'authentifier.
Scenario normal	
1.Envoyer une demande d'ajout :	<ul style="list-style-type: none"> <li>— L'utilisateur sélectionne le contact et lui envoi une invitation .</li> </ul>
2.Accepter une demande d'ajout :	<ul style="list-style-type: none"> <li>— Le système notifie l'utilisateur qu'il a une nouvelle demande d'ajout.</li> <li>— L'utilisateur consulte sa liste d'invitation et accepte/refuse la demande d'ajout.</li> </ul>
3.Retirer un contact :	<ul style="list-style-type: none"> <li>— L'utilisateur consulte la liste de ses contacts.</li> <li>— Il sélectionne le contact à retirer.</li> </ul>
Post-Condition	Liste de contact géré
Exception	Pas d'accès internet .

## Cas d'utilisation : Poster des annonces

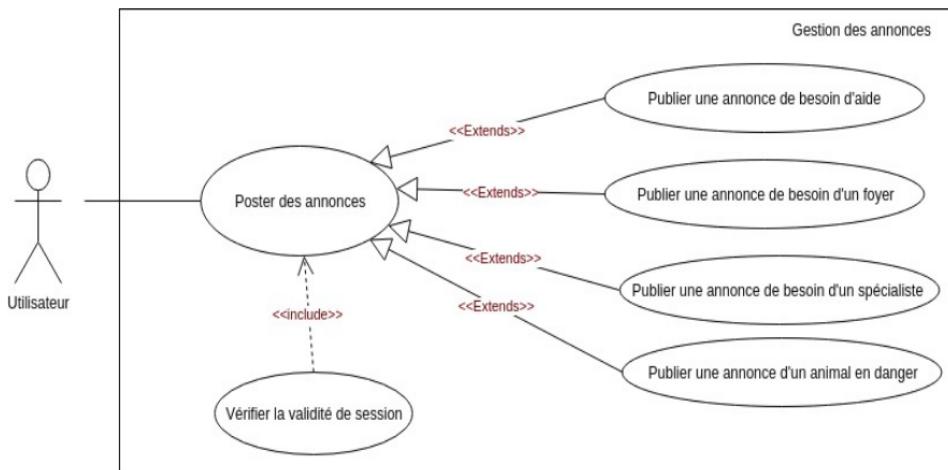


FIGURE 2.4 – Diagramme de cas d'utilisation « Poster des annonces »

Titre	Poster des annonces
Description	L'application permet aux utilisateurs de publier des annonces d'animaux en cas d'urgences de 4 types (annonce de besoin d'aide A3, annonce de besoin de foyer A2, annonce de besoin d'un spécialiste A1, annonce d'un animal en danger A0).
Acteur	Utilisateur
Pré-Condition	Vérification de la validité de session et la réputation de l'utilisateur en cas d'une annonce du type A1 ou A0 et l'activation du GPS
Scenario normal	
Déclarer une annonce	<p>L'utilisateur doit choisir le type d'annonce, puis accéder à l'interface de déclaration d'annonce où il doit introduire une description et il peut aussi ajouter des images et il précise sa localisation.</p> <p>Dans le cas où la réputation de l'utilisateur est supérieure à 30 points , il peut poster une annonce de n'importe quel type.</p> <p>Dans le cas où la réputation de l'utilisateur est inférieure à 30 points , il a le droit de poster des annonces seulement de type A3 ou A2 .</p> <p>L'avantage des annonces A0 et A1 est que la notification de ces annonces atteint les utilisateurs dans une zone géographique de 8 Km pour A1 et 10 Km pour A0 .</p>
Post-Condition	Announce publiée
Exception	<ul style="list-style-type: none"> <li>• Exception 1 :Champs obligatoires incomplets</li> <li>• Exception 2 : Désactivation du GPS</li> <li>• Exception 3 : Pas d'accès Internet .</li> </ul>

## Cas d'utilisation : Gérer fil d'accueil

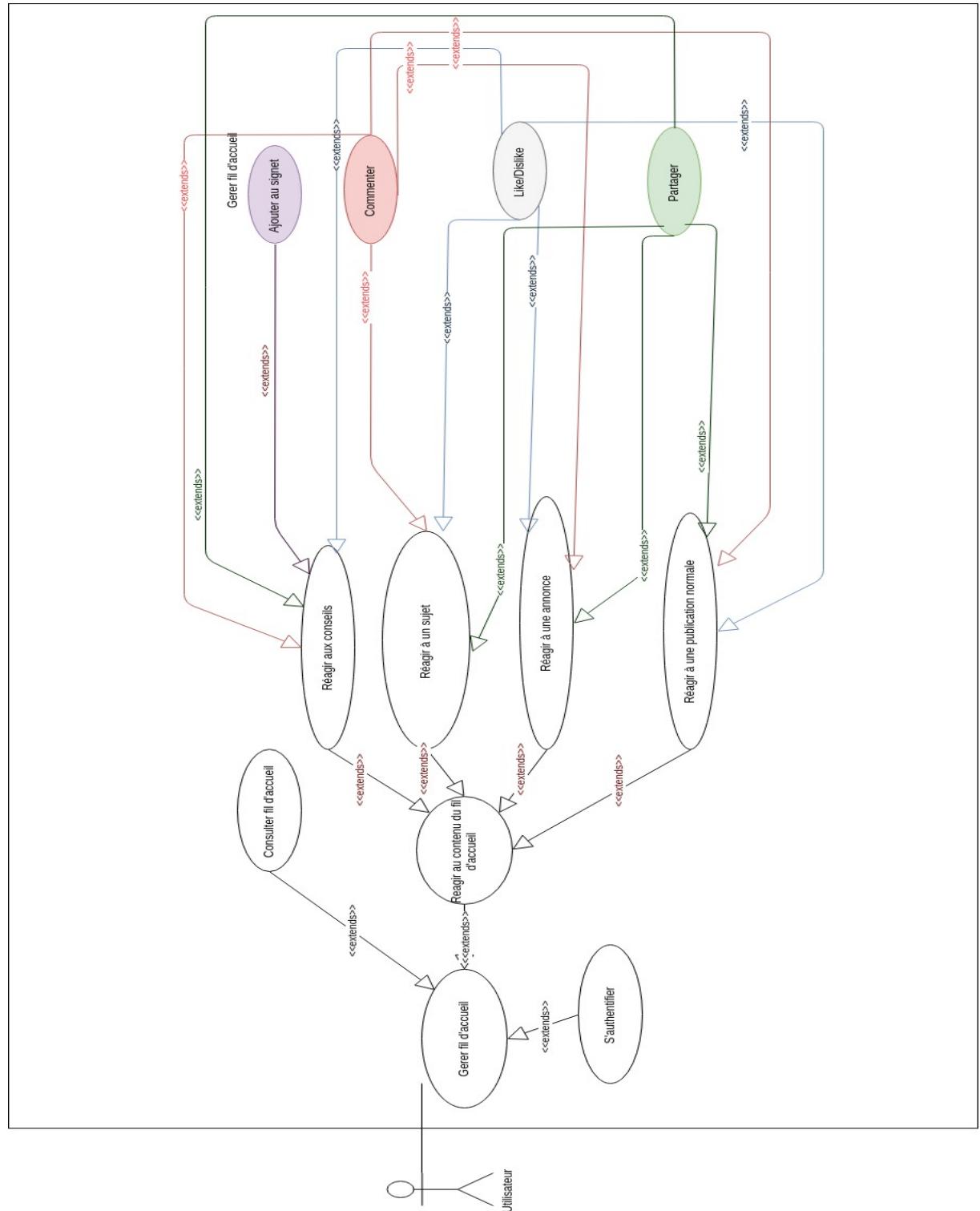


FIGURE 2.5 – Diagramme de cas d'utilisation « Gérer fil d'accueil »

Titre	Gérer il d'accueil
Description	L'application permet aux Utilisateurs de mettre à jour leur fil d'actualité ,de réagir à des publications, de les commenter , de les partager
Acteur	Utilisateur
Pré-Condition	S'authentifier
1- Consulter fil d'accueil	L'utilisateur peut consulter son fil d'accueil et visualiser les publications partagé par ses contact et les annonces à proximité ainsi que les sujet de discussion partagé par l'administrateur
2- Reagir au contenu du fil d'accueil	<ul style="list-style-type: none"> <li>— Réagir au conseils</li> <li>— J'aime/Je n'aime pas</li> <li>— Commenter</li> <li>— Partager</li> <li>— Ajouter au signet</li> <li>— Réagir à un sujet</li> <li>— J'aime / Je n'aime pas</li> <li>— Commenter</li> <li>— Réagir à une annonce</li> <li>— J'aime/Je n'aime pas</li> <li>— Commenter</li> <li>— Partager</li> <li>— Réagir à une publication normale</li> <li>— J'aime/Je n'aime pas</li> <li>— Commenter</li> <li>— Partager</li> </ul>
Post-Condition	Fil d'actualité Géré
Exception	Pas d'accès Internet .

## Cas d'utilisation : Contrôler les annonces

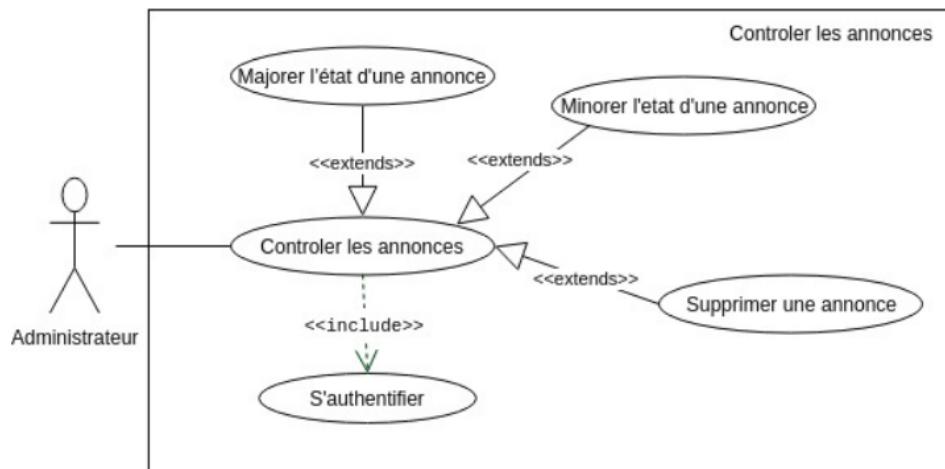


FIGURE 2.6 – Diagramme de cas d'utilisation « Contrôler les annonces »

Titre	Contrôler les annonces
Description	L'administrateur a la possibilité de contrôler les annonces des utilisateurs
Acteur	Administrateur .
Pré-Condition	<ul style="list-style-type: none"> <li>— S'authentifier.</li> <li>— Existence de l'annonce</li> </ul>
Scenario normal	
<p>L'administrateur peut consulter les annonces publié par les utilisateurs , ainsi il a la possibilité de :</p> <ol style="list-style-type: none"> <li>1- augmenter l'état d'urgence d'une annonce .</li> <li>2- diminuer l'état d'urgence d'une annonce .</li> <li>3- supprimer une annonce lorsqu'elle sont contenu est indésirable .</li> </ol>	
Post-Condition	Annonce contrôlé / modifié
Exception	Pas d'accès internet.

## Cas d'utilisation : Poster sujet de discussion

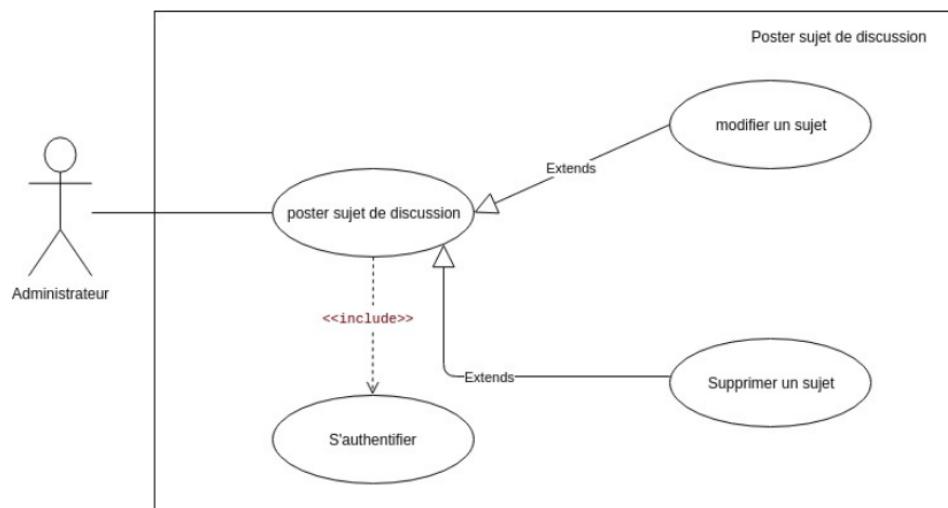


FIGURE 2.7 – Diagramme de cas d'utilisation «Poster sujet de discussion »

Titre	Poster sujet de discussion	
Description	L'administrateur de l'application peut poster un sujet de discussion .Ce poste sera visible à tout les utilisateurs ou ces dernier peuvent contribuer et partager leur avis	
Acteur	Administrateur	
Pré-Condition	S'authentifier.	
Scenario normal		
1.Poster un sujet :	En se situant sur son espace d'administration , l'administrateur pour créer un sujet de discussion et le partager pour le rendre visible aux utilisateur	
2.Modifier/supprimer un sujet :	L'administrateur peut modifier le contenu du sujet ou même le supprimé	
Post-Condition	Sujet publié , modifié ou supprimer	
Exception	<ul style="list-style-type: none"> <li>— Pas d'accès internet.</li> <li>— Champs vide</li> <li>— Pas d'accès internet</li> </ul>	

## 2.5 Spécification préliminaire des maquettes

### 2.5.1 Front Office

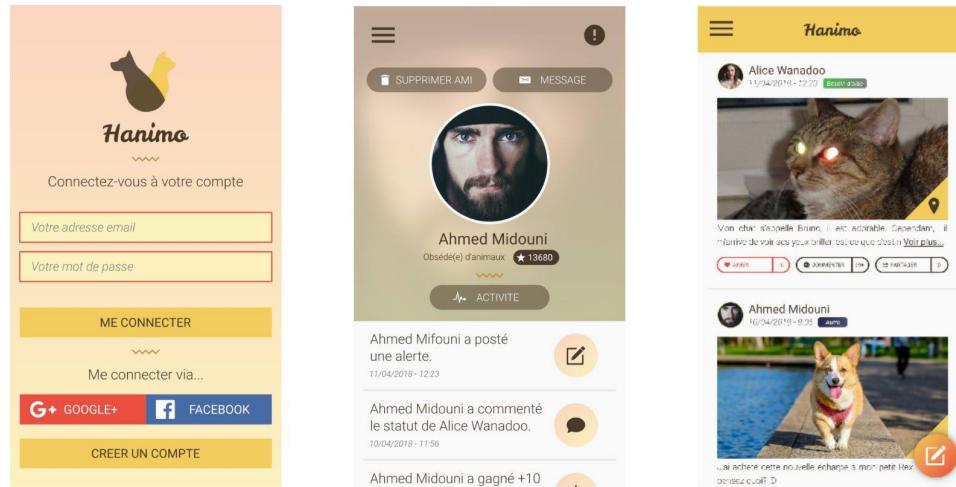


FIGURE 2.8 – Maquette de l’application pour la Front Office

- La première image représente l’interface de connexion à l’application
- La deuxième image représente le profil d’un utilisateur ami(e)
- La troisième image représente l’interface d’accueil de l’application

### 2.5.2 Back Office

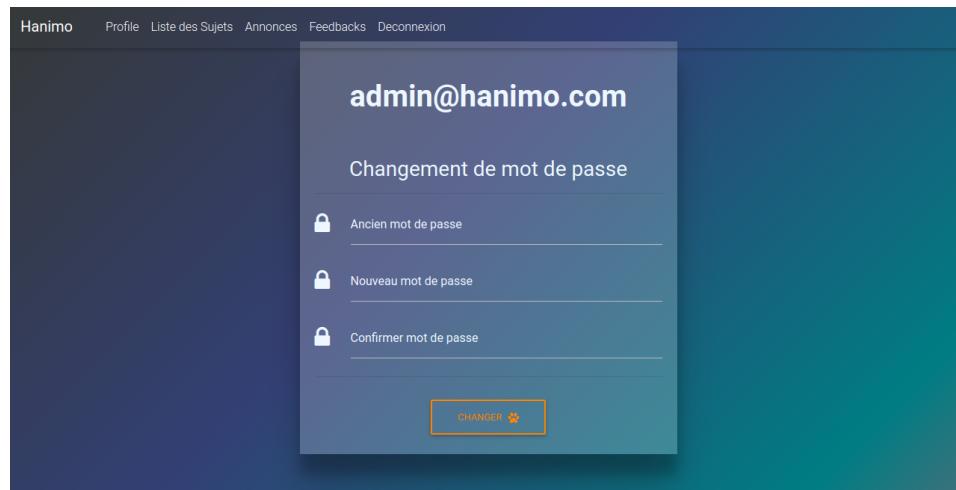


FIGURE 2.9 – Gestion de profil d’un administrateur

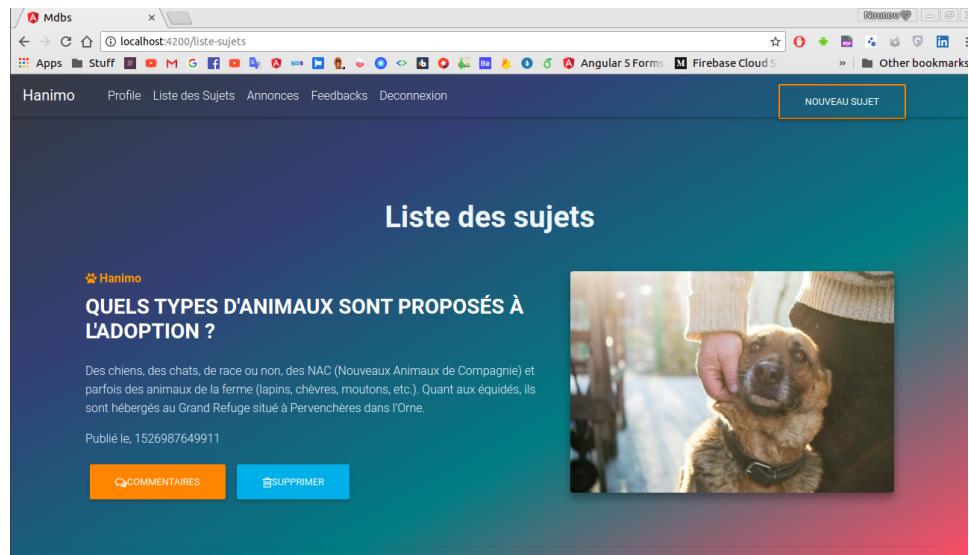


FIGURE 2.10 – Visualisation des listes des sujets

## 2.6 Conclusion

Dans ce chapitre nous avons détaillé les besoins de notre application en citant les besoins fonctionnels et non fonctionnels qui sont indispensables pour mieux faciliter le travail à réaliser. Ensuite nous avons étudié les technologies les plus connus pour choisir ionic 3 pour le développement de l'application mobile , Angular 5 comme système de front end du back office et Firebase comme un système de back end de gestion et de manipulation de données. Ensuite nous avons mis en œuvre le diagramme de cas d'utilisation et la conception préliminaire des maquettes. Nous pouvons alors entamer la phase de conception du système.

# Chapitre 3

# Conception

## 3.1 Introduction

Dans ce chapitre , nous passons à la conception qui est la phase la plus importante dans le cycle de développement d'une application pour produire une solution de haute qualité. Dans laquelle, on présentera des différents diagrammes UML. Pour ce faire , nous commençons par présenter l'architecture global de notre solution ce qui nous permettra de détailler nos choix conceptuels à travers les diagrammes appropriés .

## 3.2 Architecture 2-tiers Firebase

Définir un modèle architecturale n'est pas une tâche simple car le modèle choisi doit être compatible avec les technologies à utiliser, nous avons décidé d'adopter l'architecture 2-tiers Firebase, qui nous permettra de définir notre patron de conception :

**L'architecture 2-tiers Firebase** ressemble beaucoup à l'architecture 3 tiers traditionnelle ( client , serveur , base de données ) sauf que la différence est l'absence de la couche serveur . En effet , on ne peut pas dire que la couche serveur est totalement absente car le back end Firebase offre des services rendant cette architecture "Servless" d'une manière abstraite puisque les traitements sur les données de notre application s'effectue à travers des échanges "realtime " de façon transparente entre le client (Front End) et la base de données comme indiqué sur la figure suivante :



FIGURE 3.1 – L'architecture 2-tiers Firebase

### 3.3 Architecture MVC(Modèle-vue-contrôleur)

Modèle-vue-contrôleur ou MVC est un motif d'architecture logicielle destiné aux interfaces graphiques . Le motif est composé de trois types de modules ayant trois responsabilités différentes : les modèles, les vues et les contrôleurs.

#### — Modèle

Élément qui contient les données ainsi que de 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.

#### — 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.

#### — Contrôleur

Module qui traite les actions de l'utilisateur, modifie les données du modèle et de la vue

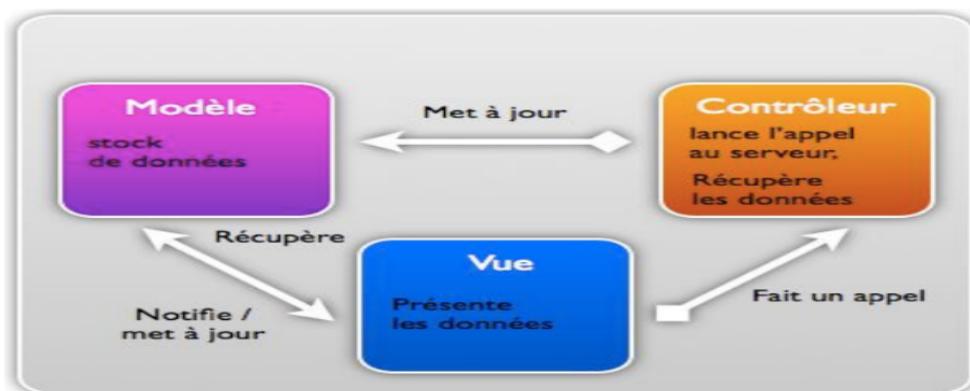


FIGURE 3.2 – Model-View-Controller (MVC)

## 3.4 Conception détaillé

### 3.4.1 Vue de kruchten

Le modèle « 4+1 » vues, dit de Kruchten, d'un système informatique permet d'organiser la description du système en plusieurs vues complémentaires, chacune présentant le système selon un point de vue différent. L'utilisation de vues permet de traiter séparément les intérêts des divers groupes d'intervenants et ainsi de mieux séparer les préoccupations fonctionnelles et les préoccupations extra fonctionnelles .

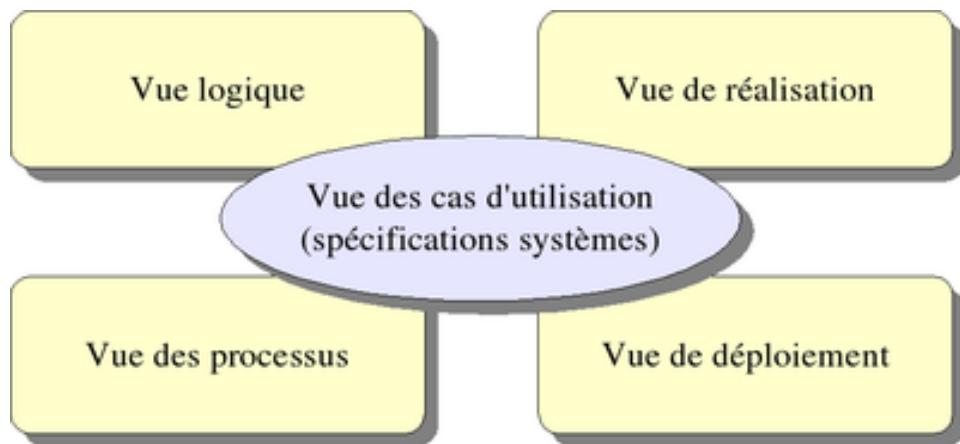


FIGURE 3.3 – Vue de Kruchten

#### Vue de cas d'utilisation

Un cas d'utilisation est défini comme un ensemble de scénarios d'utilisation, chaque scénario représentant une séquence d'interaction des utilisateurs (acteurs) avec le système. L'intérêt des cas d'utilisation est de piloter l'analyse par les exigences des utilisateurs. Ceux-ci se sentent concernés car ils peuvent facilement comprendre les cas d'utilisation qui les concernent. Cette méthode permet donc d'aider à formaliser les véritables besoins et attentes des utilisateurs. cette vue présente le :

- **Diagramme de cas d'utilisation**
- **Diagramme de package**

#### Vue logique

La vue logique constitue la principale description architecturale d'un système informatique et beaucoup de petits projets se contentent de cette seule vue. La vue logique permet d'identifier les différentes éléments et mécanismes

du système à réaliser. cette vue présente le :

- **Diagramme de classes**
- **Diagramme d'objet**

### **La vue processus**

La vue des processus décrit les interactions entre les différents processus. Cette vue permet de vérifier le respect des contraintes de fiabilité et des performances des systèmes multitâches. cette vue présente le :

- **Diagramme de séquence**

Les diagrammes de séquences permettent de représenter des collaborations entre objets selon un point de vue temporel, on y met l'accent sur la chronologie des envois de messages. Il permet de décrire comment les éléments du système interagissent entre eux et avec les acteurs en se basant sur deux dimensions.

Dimension vertical (axe de temps) :l'ordre d'envoi d'un message est déterminé par sa position sur l'axe vertical du diagramme, le temps s'écoule de haut en bas de cet axe. Dimension horizontal (les objets et les acteurs) :l'ordre de disposition des objets sur l'axe horizontal est sans importance.

Les éléments de base de ce diagramme sont :

- Les objets interagissent dans un scénario.
- Les différents types des messages envoyés :une description est écrite au dessus de la flèche.
- Représentation graphique de la ligne de vie de chaque objet.

- **Diagramme d'activité**

C'est un Diagramme associé à un objet particulier ou à un ensemble d'objets, qui illustre les flux entre les activités et les actions. Il permet de représenter graphiquement le déroulement d'un cas d'utilisation. Composition d'un diagramme d'activités Le diagramme d'activité se compose des éléments suivants :

**Une activité** représente une exécution d'un mécanisme , autrement dit, un déroulement d'étapes séquentielles.

**Une transition** qui représente Le passage d'une activité vers une autre. Cette transition peut être automatique, qui se déclenche par la fin d'une

activité, provoquent le début immédiat d'une autre ou conditionnelle ,qui ne se déclenche qu'après la satisfaction de la condition qu'on appelle aussi garde.

**Les gardes** qui représentent la condition de passage d'une activité à une autre dans les transitions conditionnelles ils sont symbolisés par des losanges.

### **La vue de réalisation**

La vue de réalisation permet de visualiser l'organisation des composants dans l'environnement de développement. Elle permet également de gérer la configuration. cette vue présente le :

- **Diagramme de composants**

Le diagramme de composants définit l'architecture logicielle du système dans un environnement de développement donné. Il est issu de la conception et permet de représenter le système et les sous-systèmes du modèle physique de l'architecture logicielle à réaliser. Un système ou un sous-système définit un espace de visibilité et regroupe des classes.

### **La vue de déploiement**

La vue de déploiement représente le système dans son environnement d'exécution. Elle traite des contraintes géographiques (distribution des processeurs dans l'espace), des contraintes de bandes passantes, du temps de réponse et des performances du système ainsi que de la tolérance aux fautes et aux pannes. Cette vue est fort utile pour l'installation et la maintenance régulière du système. cette vue présente le :

- **Diagramme de déploiement**

Les diagrammes de déploiement sont utilisés pour représenter l'architecture physique d'un système. Ils montrent la distribution des composants logiciels sur la base d'unités d'exécution (les nœuds).

### 3.4.2 Conception de cas d'utilisation "Gérer profile"

#### Vue processus :Diagramme de séquence

Titre	Gérer le profil
Description	L'utilisateur a la possibilité de gérer son profil
Acteur	Utilisateur .
Pré-Condition	S'authentifier.
Scénario normal	
1- Modifier ses informations personnelles 2- Gérer ses publications 3- Gérer Compte	<p>L'utilisateur peut modifier ses informations personnel tel que numéro de téléphone , langue , adresse ou description .</p> <ul style="list-style-type: none"> <li>— Créer conseil</li> <li>— Gérer contenu privée</li> <li>— Gérer des publications normales</li>   <li>— Modifier mot de passe</li> <li>— Se déconnecter</li> </ul>
Post-Condition	Profil géré
Exception	Pas d'accès internet .

TABLE 3.1 – Description textuelle du cas d'utilisation « Gérer profile »  
La figure suivante représente le diagramme de séquence «Gérer profile» où l'utilisateur peut gérer son contenu privé ,son compte et ses publications et modifier ses informations personnels.

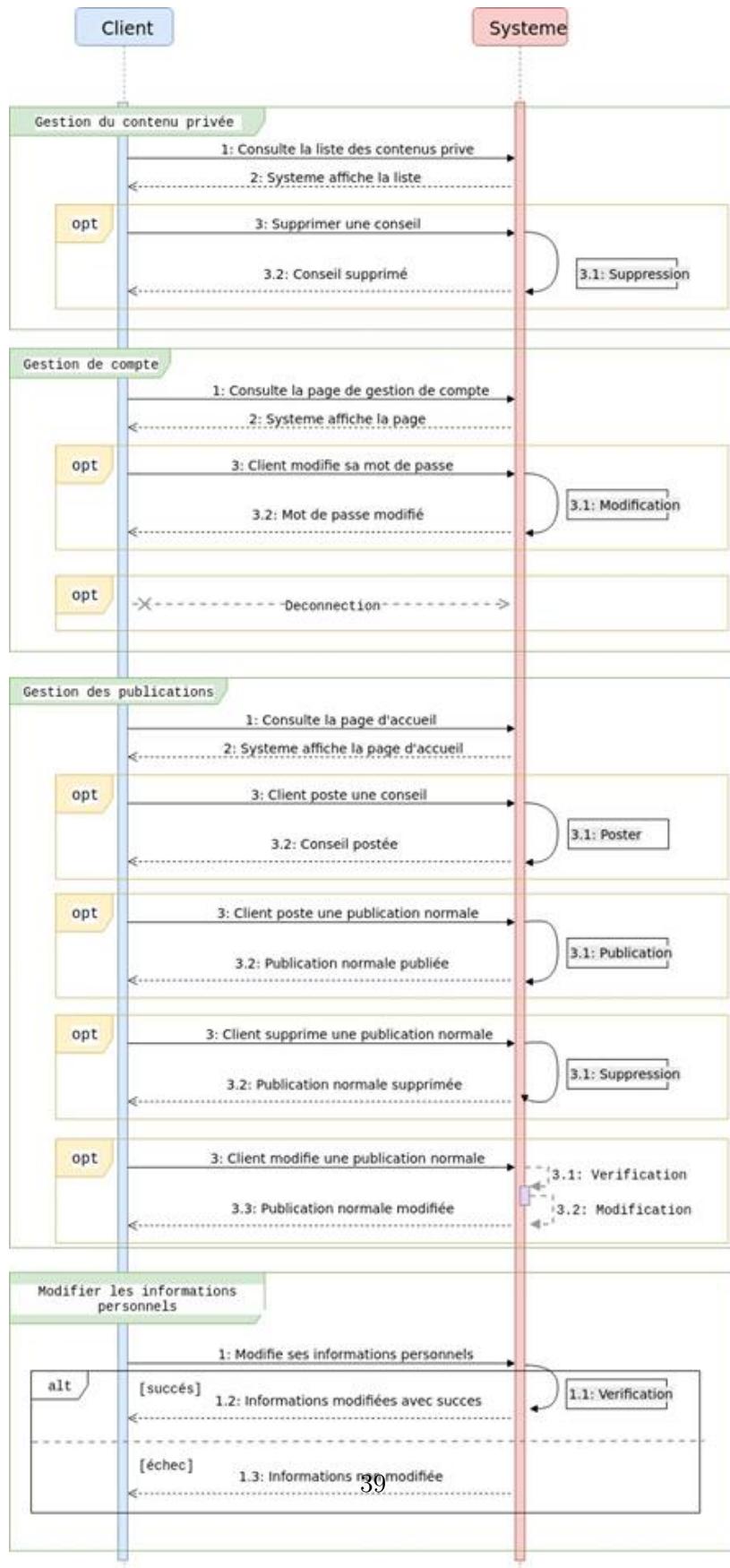


FIGURE 3.4 – Diagramme de séquence «Gérer profile»

### Vue logique :Diagramme de classes

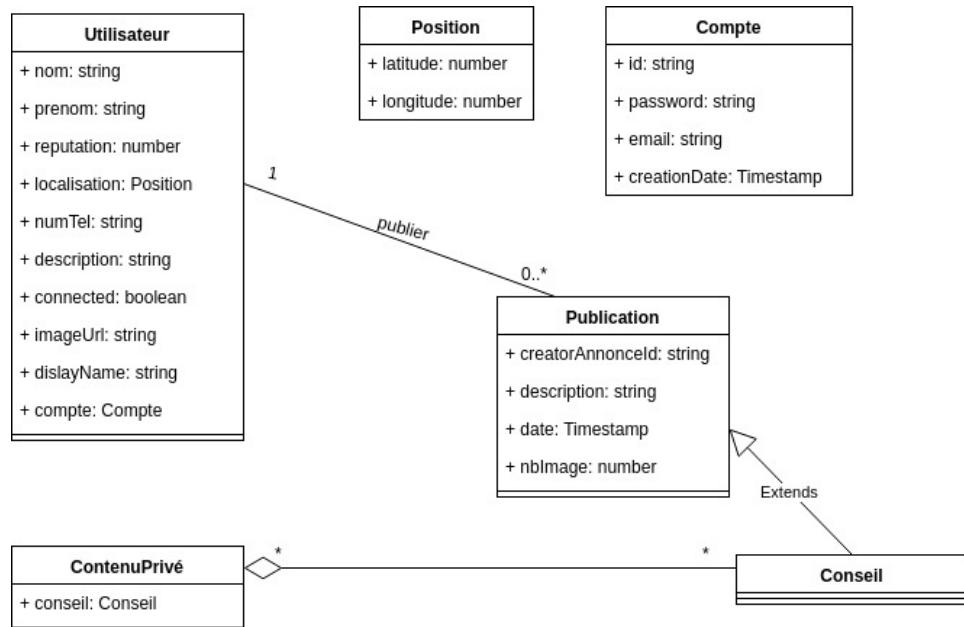


FIGURE 3.5 – Diagramme de classes «Gérer profile»

### Vue de réalisation :Diagramme de composant

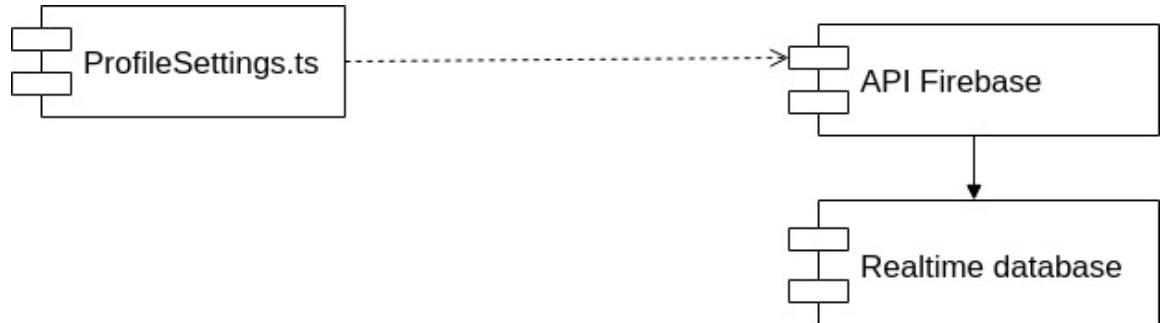


FIGURE 3.6 – Diagramme de composant «Gérer profile»

## Vue de déploiement :Diagramme de déploiement

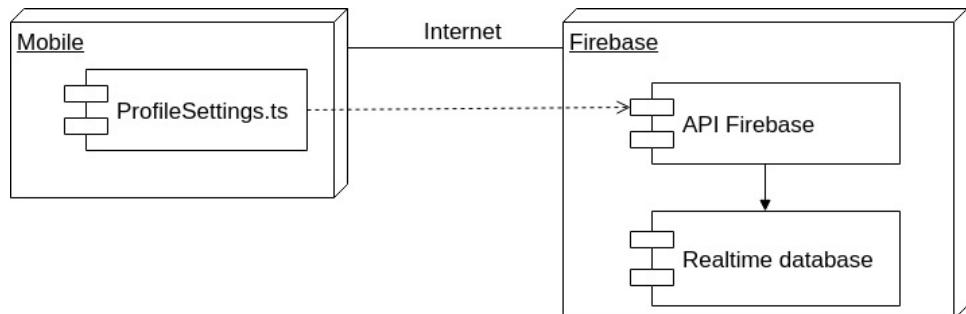


FIGURE 3.7 – Diagramme de déploiement «Gérer profile»

### 3.4.3 Conception de cas d'utilisation "Poster annonce"

#### Vue processus :Diagramme de séquence

Titre	Poster des annonces
Description	L'application permet aux utilisateurs de publier des annonces d'animaux en cas d'urgences de 4 types (annonce de besoin d'aide A3, annonce de besoin de foyer A2, annonce de besoin d'un spécialiste A1, annonce d'un animal en danger A0).
Acteur	Utilisateur
Pré-Condition	Vérification de la validité de session et la réputation de l'utilisateur en cas d'une annonce du type A1 ou A0 et l'activation du GPS
Scenario normal	
Déclarer une annonce	<p>L'utilisateur doit choisir le type d'annonce, puis accéder à l'interface de déclaration d'annonce où il doit introduire une description et il peut aussi ajouter des images et il précise sa localisation.</p> <p>Dans le cas où la réputation de l'utilisateur est supérieure à 30 points , il peut poster une annonce de n'importe quel type.</p> <p>Dans le cas où la réputation de l'utilisateur est inférieur à 30 points , il a le droit de poster des annonces seulement de type A3 ou A2 .</p> <p>L'avantage des annonces A0 et A1 est que la notification de ces annonces atteint les utilisateurs dans une zone géographique de 8 Km pour A1 et 10 Km pour A0 .</p>
Post-Condition	Annonce publiée
Exception	<ul style="list-style-type: none"> <li>• Exception 1 :Champs obligatoires incomplets</li> <li>• Exception 2 : Désactivation du GPS</li> <li>• Exception 3 : Pas d'accès Internet .</li> </ul>

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

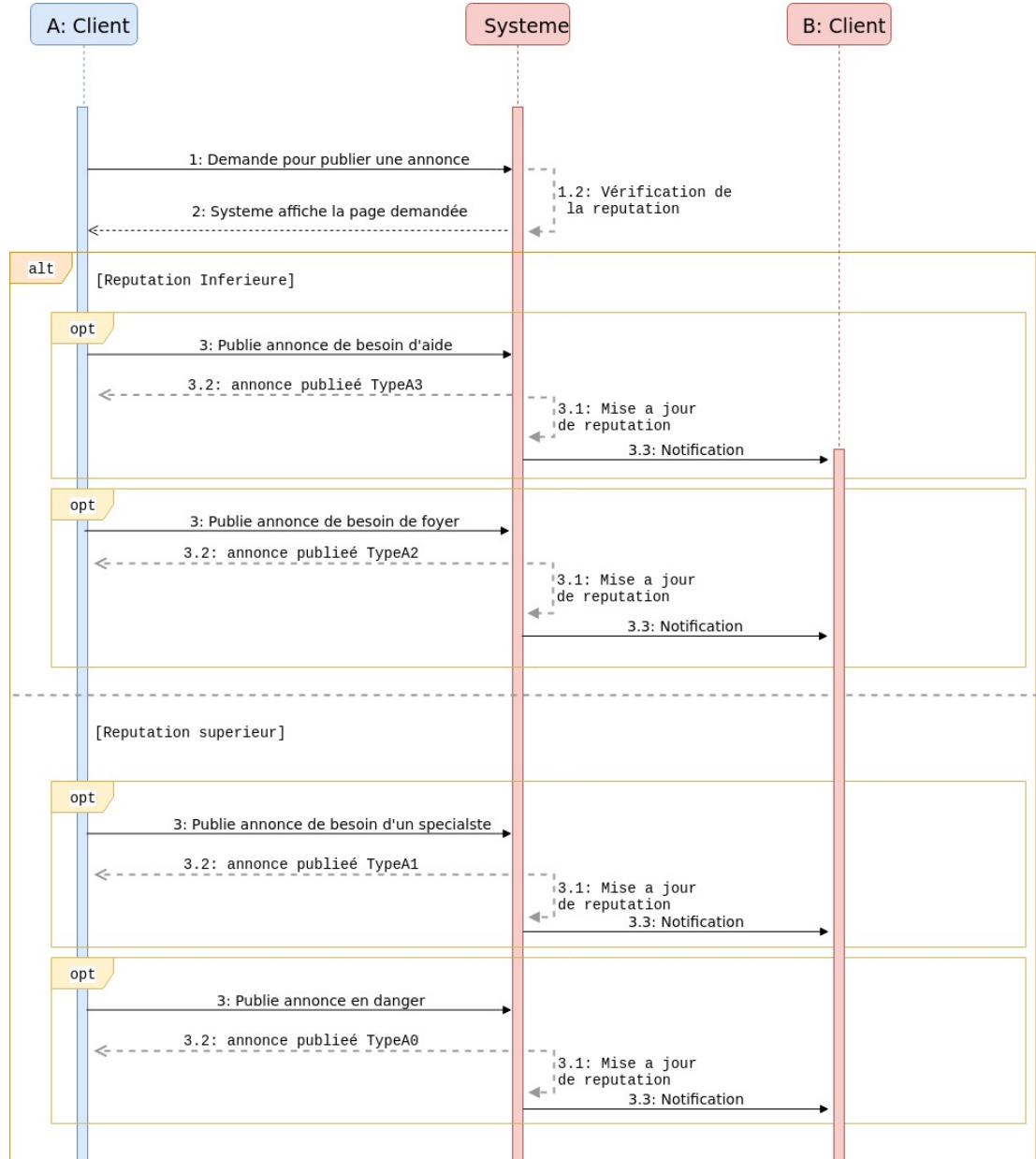


FIGURE 3.8 – Diagramme de séquence «Poster annonce»

### Vue logique :Diagramme de classes

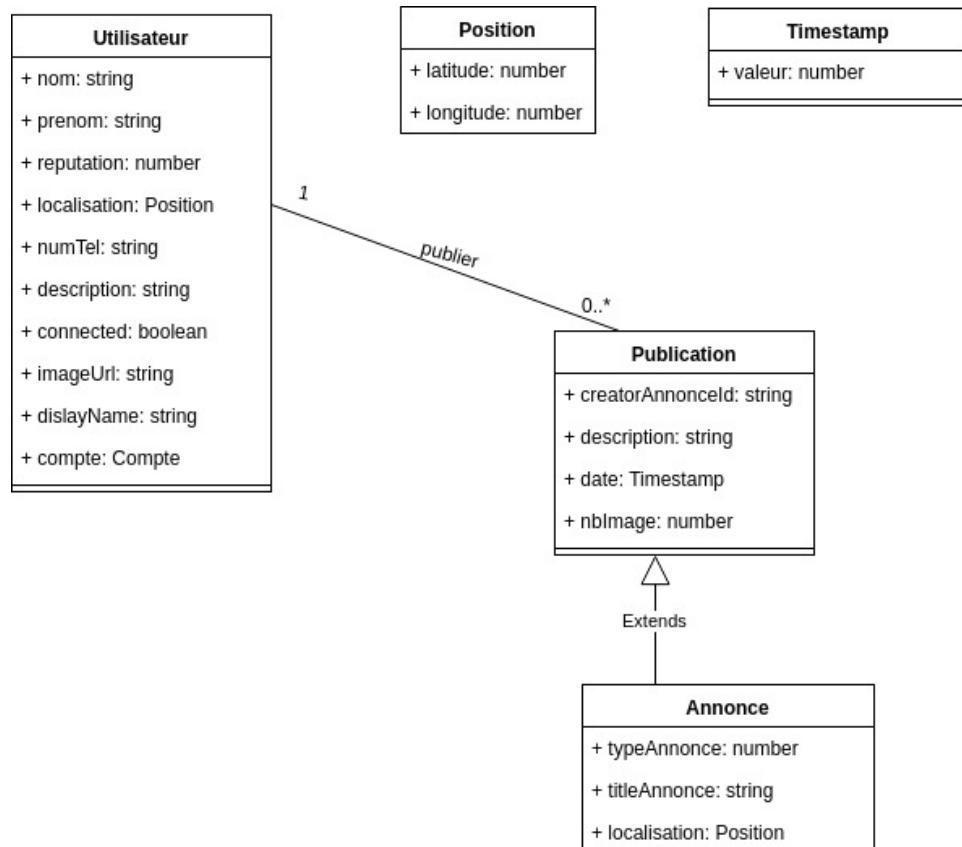


FIGURE 3.9 – Diagramme de classes «Poster annonce»

### Vue de réalisation :Diagramme de composant

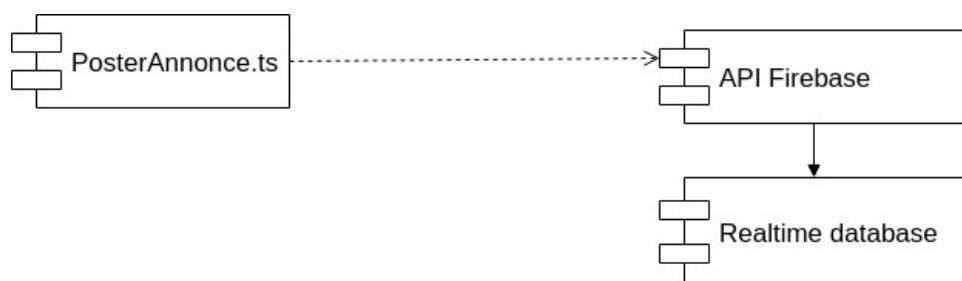


FIGURE 3.10 – Diagramme de composant «Poster annonce»

## Vue de déploiement :Diagramme de déploiement

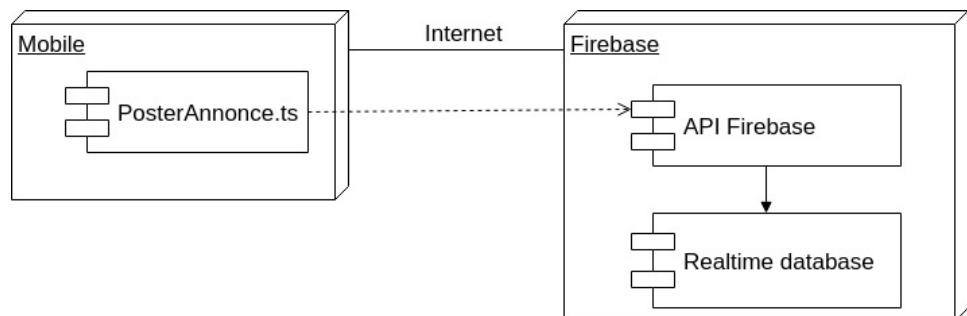


FIGURE 3.11 – Diagramme de déploiement «Poster annonce»

### 3.4.4 Conception de cas d'utilisation "Gérer fil d'accueil"

#### Vue processus :Diagramme de séquence

Titre	Gérer fil d'accueil
Description	L'application permet aux Utilisateurs de mettre à jour leur fil d'actualité ,de réagir à des publications, de les commenter , de les partager
Acteur	Utilisateur
Pré-Condition	S'authentifier
1- Consulter fil d'accueil	L'utilisateur peut consulter son fil d'accueil et visualiser les publications partagé par ses contact et les annonces à proximité ainsi que les sujet de discussion partagé par l'administrateur
2- Réagir au contenu du fil d'accueil	<ul style="list-style-type: none"> <li>— Réagir au conseils           <ul style="list-style-type: none"> <li>— J'aime/Je n'aime pas</li> <li>— Commenter</li> <li>— Partager</li> <li>— Ajouter au signet</li> </ul> </li> <li>— Réagir à un sujet           <ul style="list-style-type: none"> <li>— J'aime / Je n'aime pas</li> <li>— Commenter</li> </ul> </li> <li>— Réagir à une annonce           <ul style="list-style-type: none"> <li>— J'aime/Je n'aime pas</li> <li>— Commenter</li> <li>— Partager</li> </ul> </li> <li>— Réagir à une publication normale           <ul style="list-style-type: none"> <li>— J'aime/Je n'aime pas</li> <li>— Commenter</li> <li>— Partager</li> </ul> </li> </ul>
Post-Condition	Fil d'actualité Géré
Exception	Pas d'accès Internet .

TABLE 3.3 – Description textuelle du cas d'utilisation « Gérer fil d'accueil »

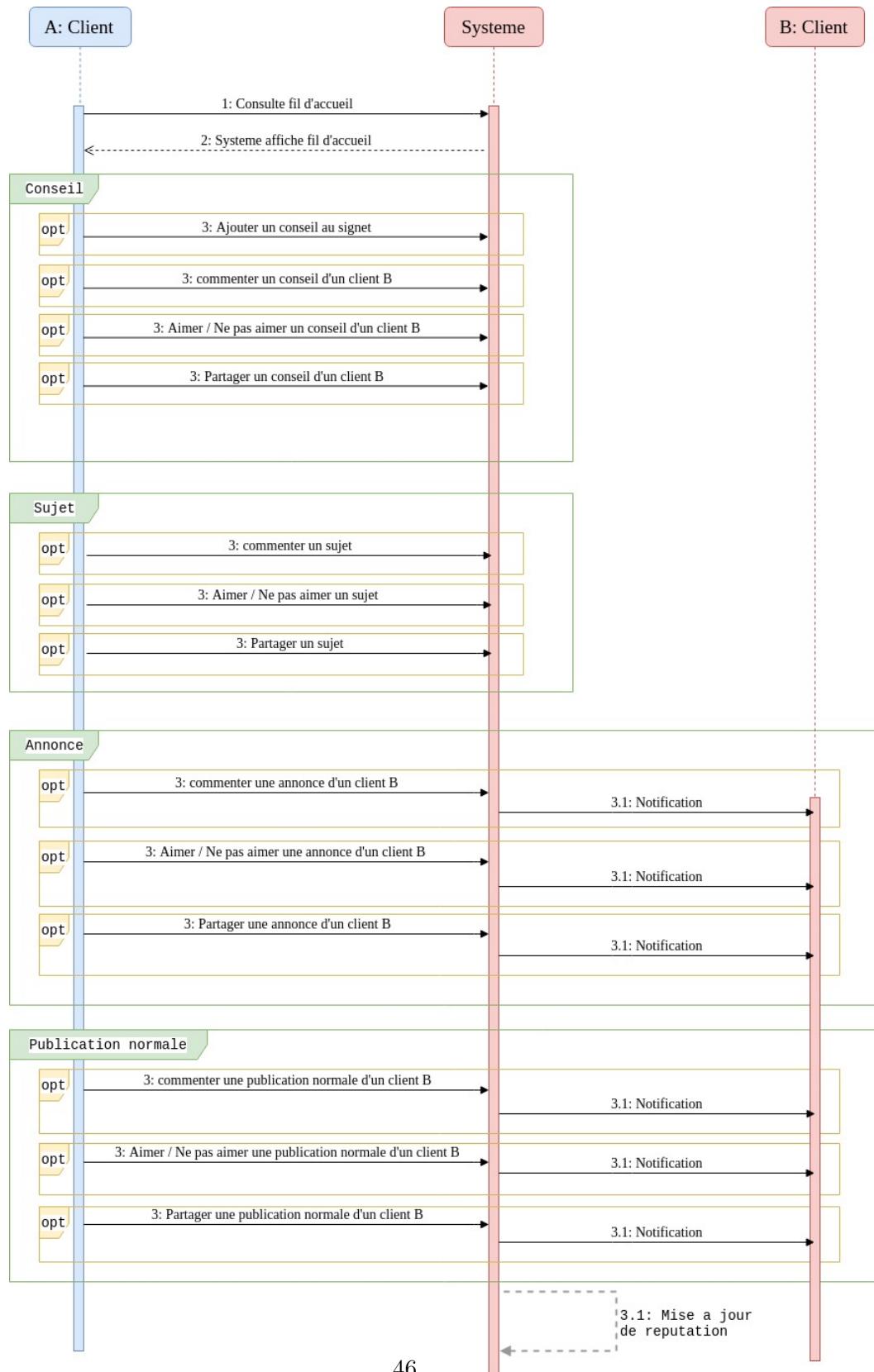


FIGURE 3.12 – Diagramme de séquence «Gérer fil d'accueil»

## Vue logique :Diagramme de classes

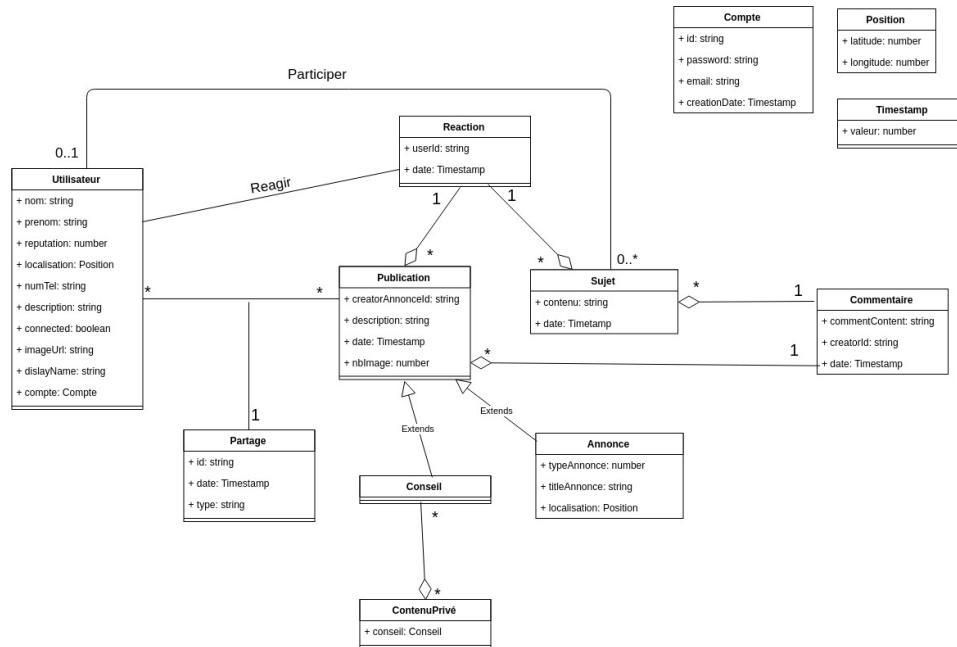


FIGURE 3.13 – Diagramme de classes «Gérer fil d'accueil»

## Vue de réalisation :Diagramme de composant

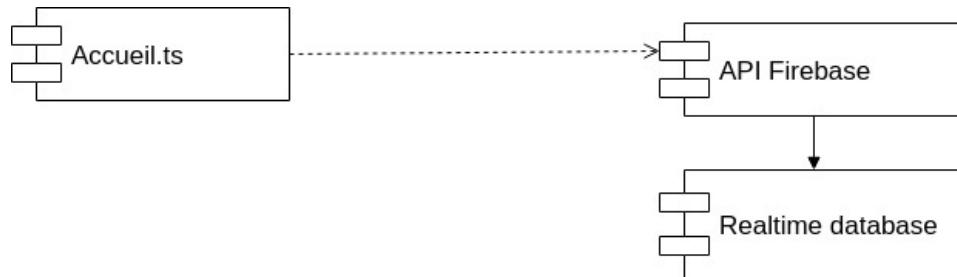


FIGURE 3.14 – Diagramme de composant «Gérer fil d'accueil»

## Vue de déploiement :Diagramme de déploiement

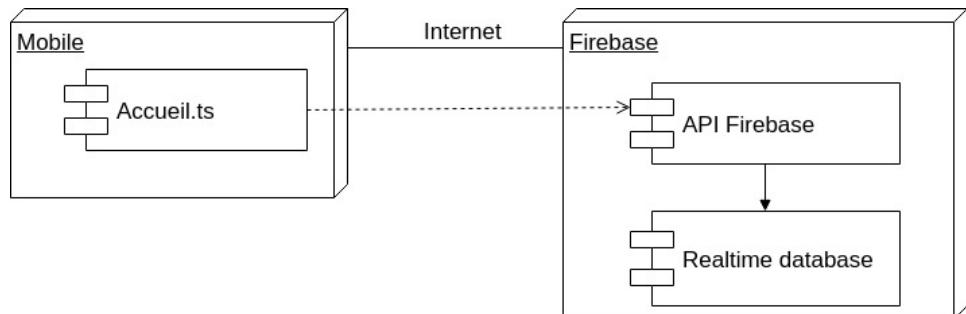


FIGURE 3.15 – Diagramme de déploiement «Gérer fil d'accueil»

### 3.4.5 Conception de cas d'utilisation "Contrôler les annonces"

#### Vue processus :Diagramme de séquence

Titre	Contrôler les annonces
Description	L'administrateur a la possibilité de contrôler les annonces des utilisateurs
Acteur	Administrateur .
Pré-Condition	<ul style="list-style-type: none"> <li>— S'authentifier.</li> <li>— Existence de l'annonce</li> </ul>
Scenario normal	
<p>L'administrateur peut consulter les annonces publiées par les utilisateurs , ainsi il a la possibilité de :</p> <ol style="list-style-type: none"> <li>1- augmenter l'état d'urgence d'une annonce .</li> <li>2- diminuer l'état d'urgence d'une annonce .</li> <li>3- supprimer une annonce lorsqu'elle est considérée indésirable .</li> </ol>	
Post-Condition	Annonce contrôlée / modifiée
Exception	Pas d'accès internet.

TABLE 3.4 – Description textuelle du cas d'utilisation « Contrôler les annonces »

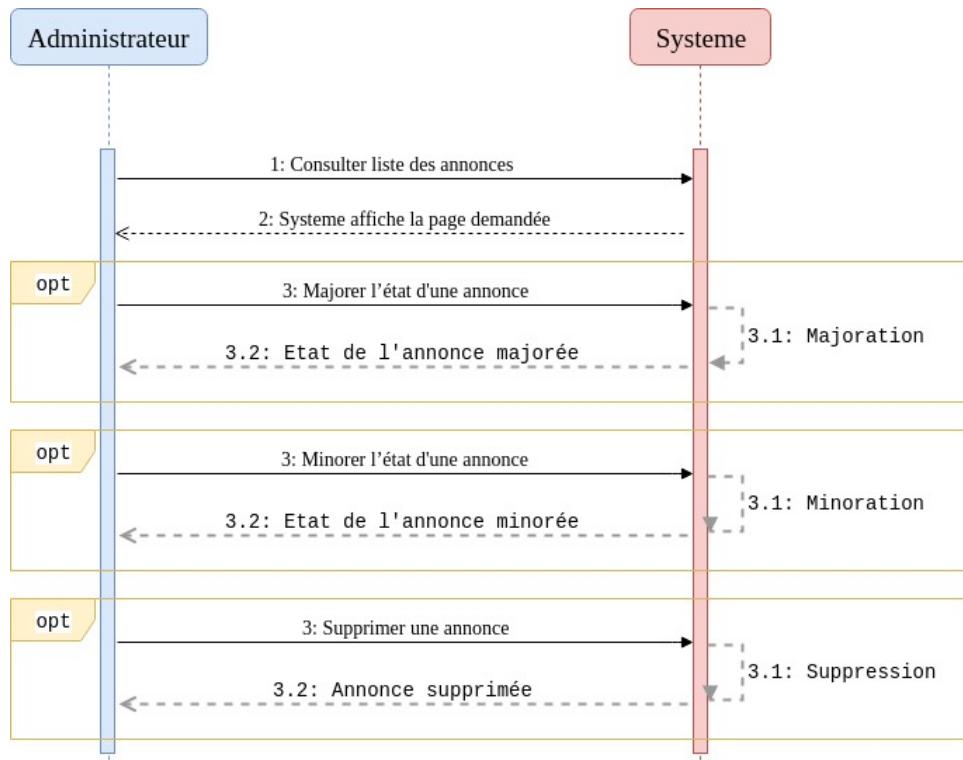


FIGURE 3.16 – Diagramme de séquence «Contrôler les annonces»

### Vue logique :Diagramme de classes

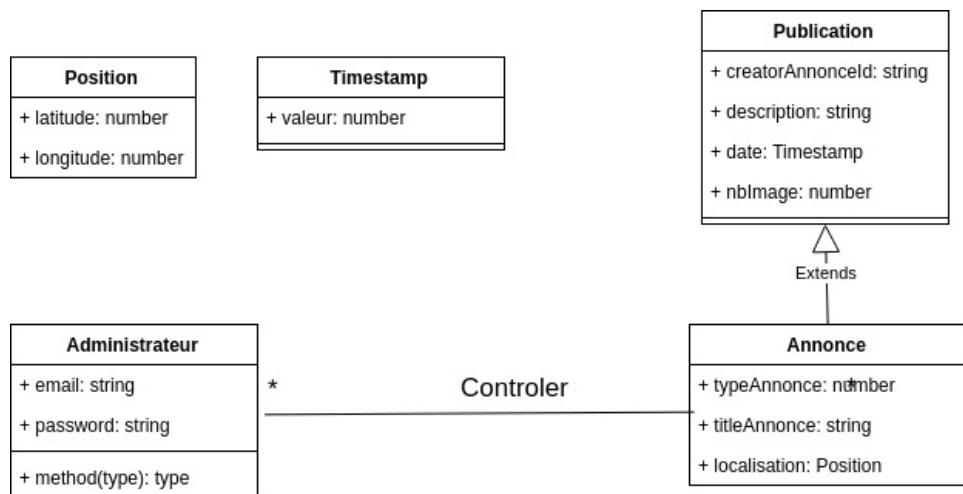


FIGURE 3.17 – Diagramme de classes «Contrôler les annonces»

### Vue de réalisation :Diagramme de composant

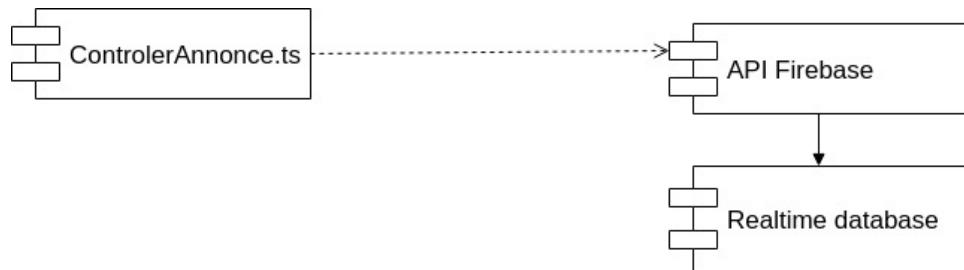


FIGURE 3.18 – Diagramme de composant «Contrôler les annonces»

### Vue de déploiement :Diagramme de déploiement

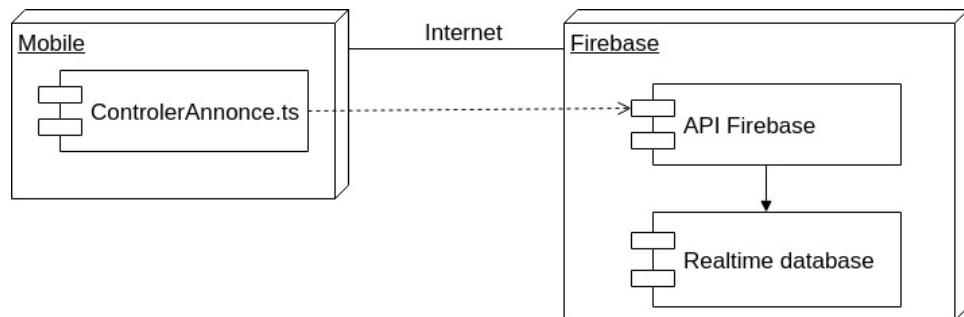


FIGURE 3.19 – Diagramme de déploiement «Contrôler les annonces»

### 3.4.6 Diagramme de classes globale

Nous présentons ici le diagramme de classes globales où nous montrons les différentes classes de notre solution ainsi que les relations entre ces classes.

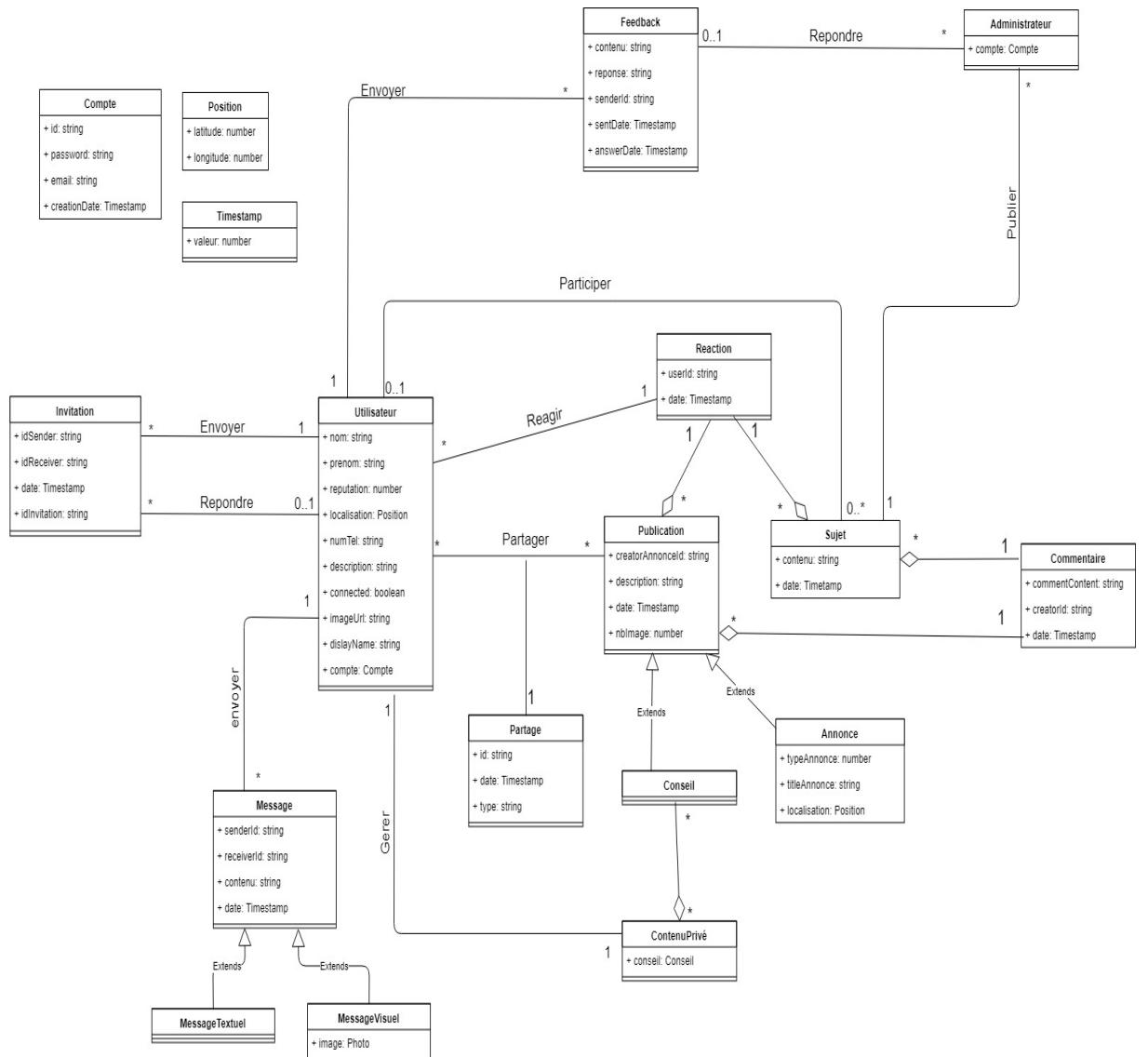


FIGURE 3.20 – Diagramme de classes globale

### 3.5 Modèle MVC proposé

La figure ci dessous est un schéma simplifié de notre application qui montre la façon d'utilisation du modèle de conception MVC.

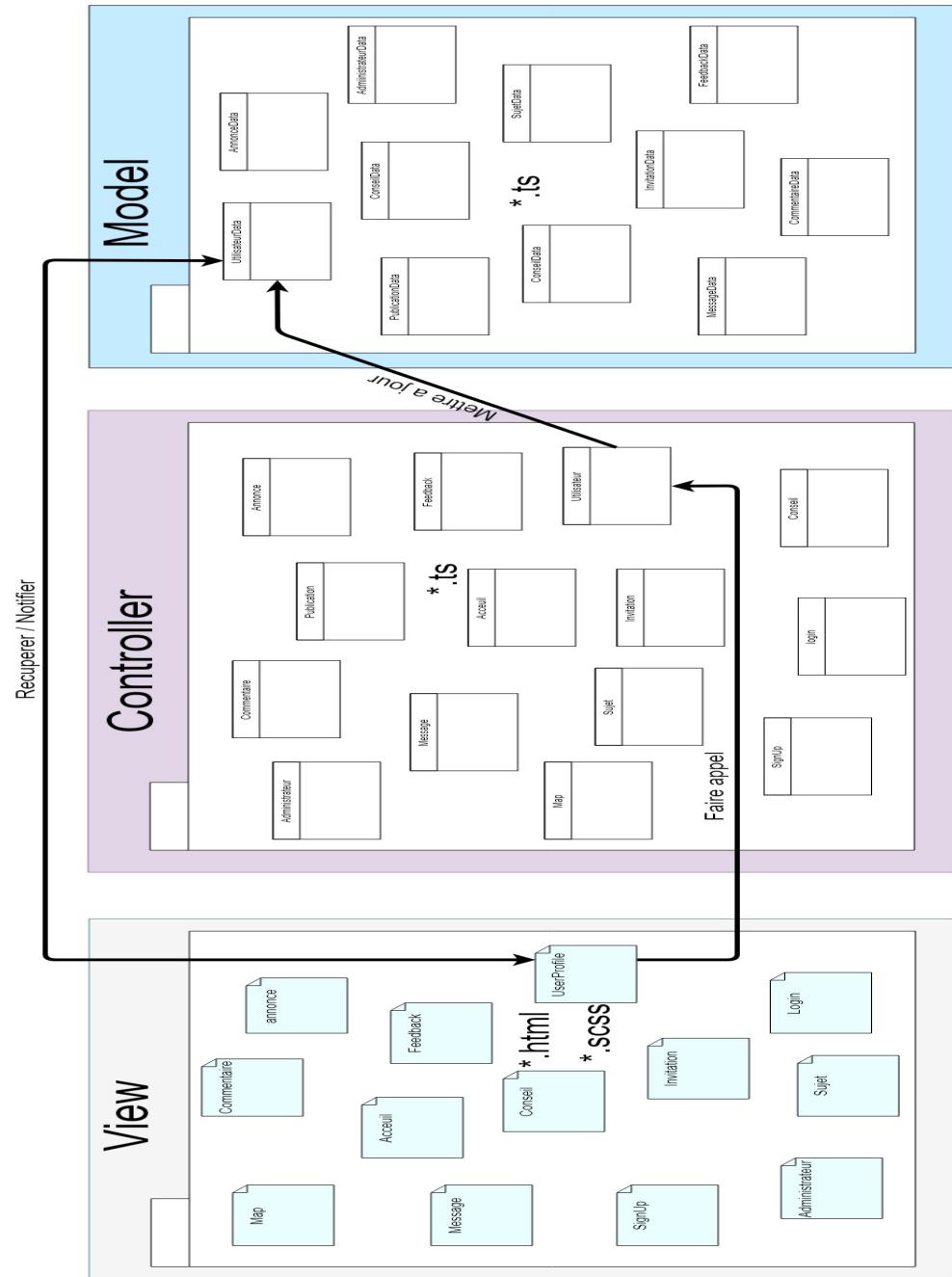


FIGURE 3.21 – Modèle MVC proposé

### **3.6 Conclusion**

Dans ce chapitre nous avons présenté le modèle architecturale qu'on a adopté qui est l'architecture 2-tiers Firebase , puis l'architecture conceptuel qui est MVC . Ensuite nous avons étudié les vue de Kruchten pour réaliser la conception détaillée et enfin le diagramme de classes global.

# Chapitre 4

## Réalisation

### 4.1 Introduction

La phase de réalisation est considérée comme la mise en œuvre concrète des éléments planifiés dans la recherche et déploiement de solutions pour satisfaire les objectifs définis.

### 4.2 Environnement de développement

#### 4.2.1 NodeJS

Node.js est une plateforme de développement Javascript avec des bibliothèques permet d'utiliser le langage JavaScript sur le serveur en dehors du navigateur. Node.js a été créé par Ryan Dahl dans le but de pouvoir créer des applications temps réel où le serveur est capable de pousser de l'information au client. C'est dans ce but qu'il utilise la bibliothèque libuv pour réaliser son modèle d'entrée sortie non bloquante.

Pourquoi utiliser Node.js ?

Node.js peut être comparé à Python, Ruby, Java, PHP. Node.js offre un environnement côté serveur et présente de nombreux intérêts :

- Logiciel libre (licence MIT)
- Performance du moteur v8 (un outil open source créé par Google qui analyse et exécute du code JavaScript très rapidement).
- Modèle non bloquant.
- Communauté très active.
- Système de paquet intégré (NPM).
- Les grandes entreprises l'utilisent.



FIGURE 4.1 – Logo NodeJs

#### 4.2.2 Ionic 3 Framework

Ionic Framework est un Open Source puissant SDK HTML5 permettant de développer des applications mobiles hybrides, native ou web mobile rapidement et facilement en langage WEB (HTML / CSS / JS). Il s'appuie sur AngularJS pour la partie application web du framework et sur Cordova pour la partie construction des applications natives déployable sur plusieurs environnements tout en gardant la même interaction et design qu'une application native.



FIGURE 4.2 – Logo Ionic

#### 4.2.3 Angular 5

Angular est un framework JavaScript qui étend le HTML pour le rendre dynamique, et permet de développer ses propres balises et attributs HTML. C'est un framework qui se veut extensible et qui pousse vers un développement structuré, en couches, le but n'étant pas d'ajouter de simples animations au DOM, mais bien d'apporter un aspect applicatif au front-office.

Les concepts qui caractérisent le plus Angular sont :

- Basé sur la logique MVC. (modèle vue contrôleur)
- Data-binding bidirectionnel (liaison entre vues, contrôleurs et le modèle)

- Injection de dépendance (charger certaines parties de l'application seulement quand c'est nécessaire)
- Routing



FIGURE 4.3 – Logo Angular

#### 4.2.4 Firebase

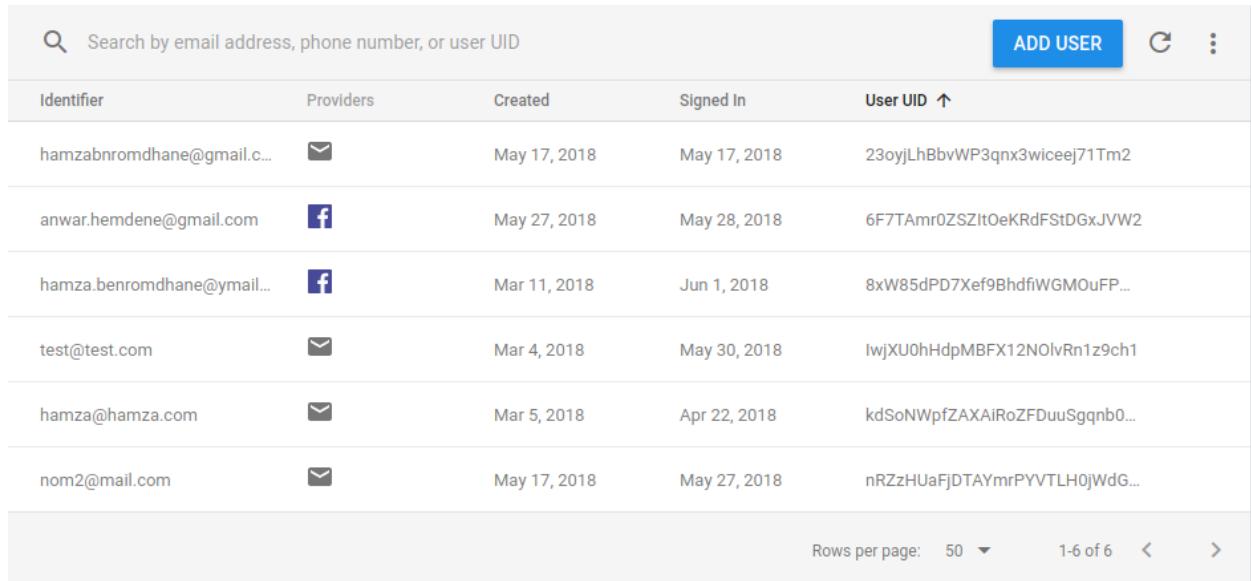
Une plateforme mobile de Google nous libère de la complexité de création et de la maintenance d'une architecture serveur performant, tout en nous garantissant une scalabilité à toute épreuve (plusieurs milliards d'utilisateurs) et une simplicité dans l'utilisation. Firebase offre plusieurs services permettant la gestion du back-end entier d'un projet.



FIGURE 4.4 – Logo Firebase

#### Authentification

Ce service de Firebase offre les fonctionnalités de gestion des comptes des utilisateurs. Il permet une variété de méthodes d'authentification, en utilisant un email et mot de passe, avec Facebook, avec Google+, Twitter et bien plus. Après l'authentification, ce service offre les méthodes nécessaires de gestion d'un compte tel que le changement de mot de passe, la récupération de compte, vérification de la validité de session, la connexion et la déconnexion.



A screenshot of the Firebase Authentication console. At the top, there is a search bar with the placeholder "Search by email address, phone number, or user UID". To the right of the search bar are three buttons: "ADD USER" (blue), a refresh icon, and a more options icon. The main area is a table with the following columns: "Identifier", "Providers", "Created", "Signed In", and "User UID ↑". The table contains six rows of user data:

Identifier	Providers	Created	Signed In	User UID ↑
hamzabnromdhane@gmail.c...		May 17, 2018	May 17, 2018	23oyjLhBbvWP3qnx3wiceej71Tm2
anwar.hemdene@gmail.com		May 27, 2018	May 28, 2018	6F7TAmr0ZSZltOeKRdFStDGxJVW2
hamza.benromdhane@ymail...		Mar 11, 2018	Jun 1, 2018	8xW85dPD7Xef9BhdflWGMouFP...
test@test.com		Mar 4, 2018	May 30, 2018	IwjXU0hHdpMBFX12N0lvRn1z9ch1
hamza@hamza.com		Mar 5, 2018	Apr 22, 2018	kdSoNWpfZAXAiRoZFDuuSgqnb0...
nom2@mail.com		May 17, 2018	May 27, 2018	nRZZHUaFjDTAYmrPYVTLHOjWdG...

At the bottom right of the table, there are buttons for "Rows per page: 50" (with a dropdown arrow), "1-6 of 6", and navigation arrows.

FIGURE 4.5 – Firebase authentication

### RealTime Database

Firebase Realtime Database est un service de base données NoSql permettant le stockage de données en ligne sous format JSON . Indépendamment de la plateforme, ces données sont synchronisé en temps réel pour tout client connecté. En NoSql les données sont stockés sous forme de collection d'objets et chaque objet peut contenir d'autre objet . La structuration de données est très flexible de façon que les objets d'une même collection peuvent avoir des attributs différents . La figure ci dessous montre une capture de notre base de données .

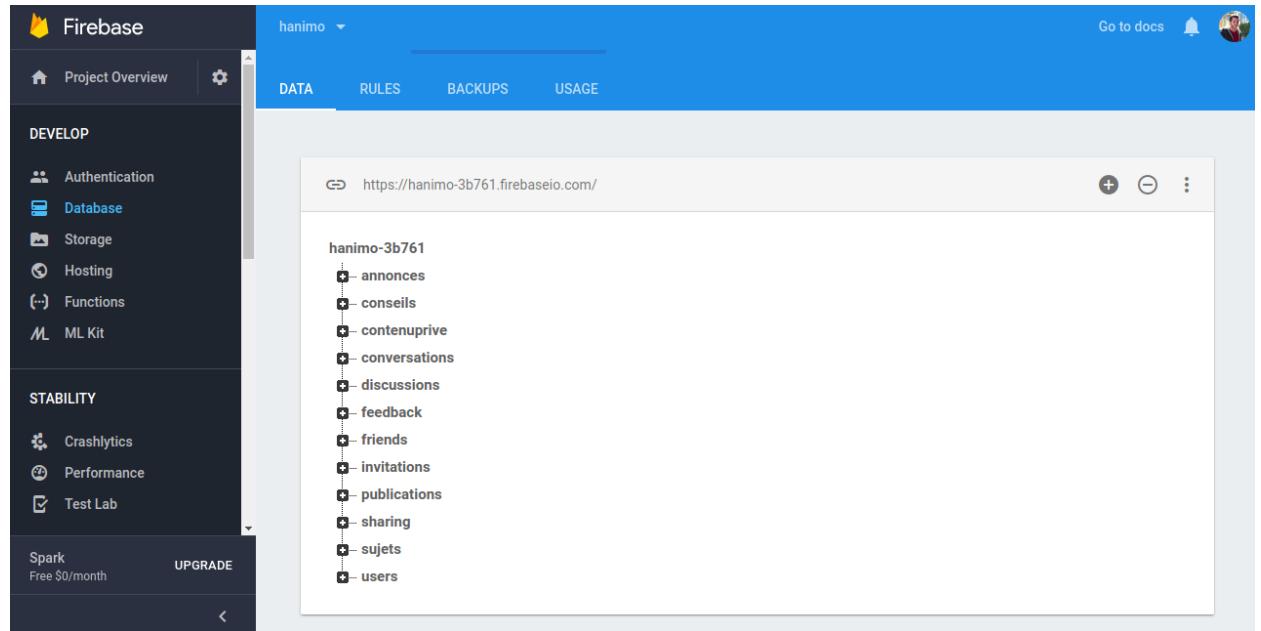


FIGURE 4.6 – Firebase Realtime Database

### Firebase Cloud Storage

C'est un puissant service de stockage d'objets en ligne. Il permet de stocker des images, vidéos, sons et tout type de fichier. Ce service offre des méthodes pour exporter les fichiers médias du client et les stocker dans le serveur ainsi que la récupération pour les visualiser plus tard.

La figure suivante présente la structuration de stockage des données média de notre application.

The screenshot shows the Firebase Cloud Storage interface. At the top, there's a warning message: 'Your security rules are defined as public, so anyone can steal, modify, or delete data in your storage bucket.' with 'LEARN MORE' and 'DISMISS' buttons. Below this, the storage structure is listed:

<input type="checkbox"/>	<input type="checkbox"/> annoncesimages/	—	Folder
<input type="checkbox"/>	<input type="checkbox"/> messagesImages/	—	Folder
<input type="checkbox"/>	<input type="checkbox"/> profileImages/	—	Folder
<input type="checkbox"/>	<input type="checkbox"/> publicationsimages/	—	Folder

FIGURE 4.7 – Firebase Cloud Storage

#### 4.2.5 Draw.io

Draw.io est une application gratuite en ligne qui permet de dessiner des diagrammes ou des organigrammes. Cet outil nous propose de concevoir toutes sortes de diagrammes et les exporter sous plusieurs formats.



FIGURE 4.8 – Logo Draw.io

#### 4.2.6 Github

Le nom GitHub est composé du mot « git » faisant référence au système de contrôle de version open-source et le mot « hub » faisant référence au réseau social bâti autour du système Git. **Git** un logiciel de gestion de version, ce qui signifie qu'il gère les modifications d'un projet sans en écraser toutes les parties.

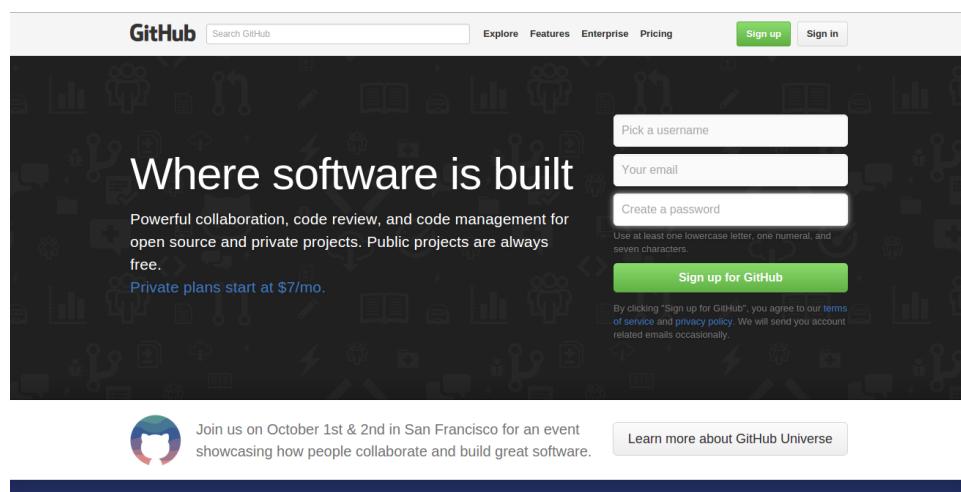


FIGURE 4.9 – Page d'accueil de Github

#### 4.2.7 Inkscape

Inkscape est un puissant éditeur libre d'images vectorielles, sous licence GNU/GPL. Il permet la création et la manipulation de graphismes en format SVG avec des fonctionnalités similaires à celles de logiciels propriétaires du secteur tels que Illustrator, CorelDraw, Freehand, Xara Xtreme etc.



FIGURE 4.10 – Page d'accueil de Inkscape

#### 4.2.8 Visual Studio Code

Visual Studio Code est un éditeur de code open-source, gratuit et multi-plateforme, développé par Microsoft. Principalement conçu pour le développement d'application avec JavaScript, TypeScript et Node.js, l'éditeur peut s'adapter à d'autres types de langages grâce à un système d'extension bien fourni.



FIGURE 4.11 – Logo Visual Studio Code

## 4.3 Présentation de la solution "Hanimo"

### 4.3.1 Front Office

#### Interface de connexion

C'est la première interface qui s'affiche, à partir de laquelle l'utilisateur peut se connecter à son compte soit en entrant ses coordonnées , adresse mail et mot de passe, soit en se connectant avec son compte Facebook ou Gmail . Ou il peut créer un compte en appuyant sur le bouton “ créer un compte ”.

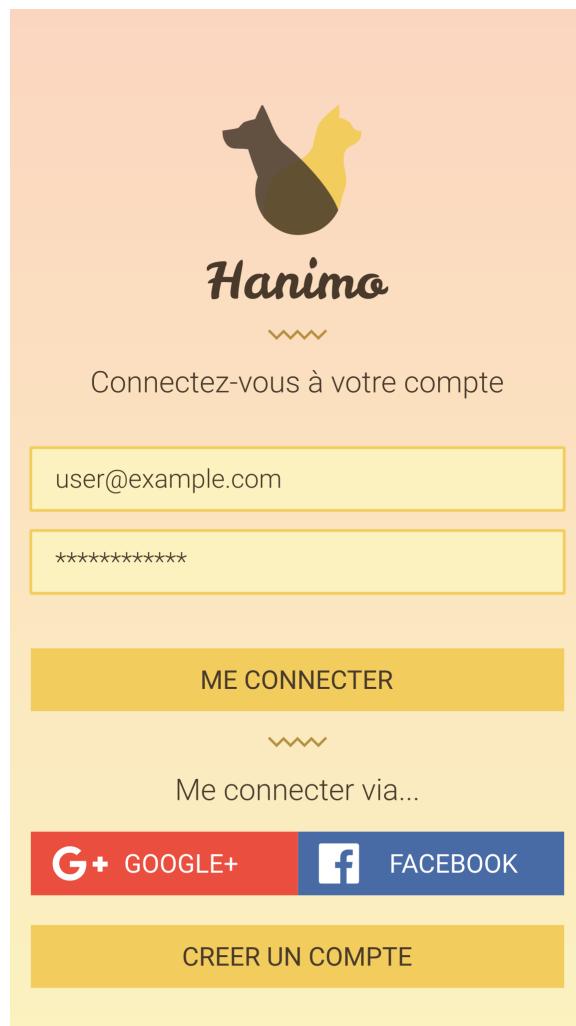


FIGURE 4.12 – Interface de connexion

## Interface d'inscription

Cette interface offre à la possibilité de créer un compte sur notre application, l'utilisateur doit saisir ses informations personnels et respecter les champs qui sont obligatoires puis valider son inscription .

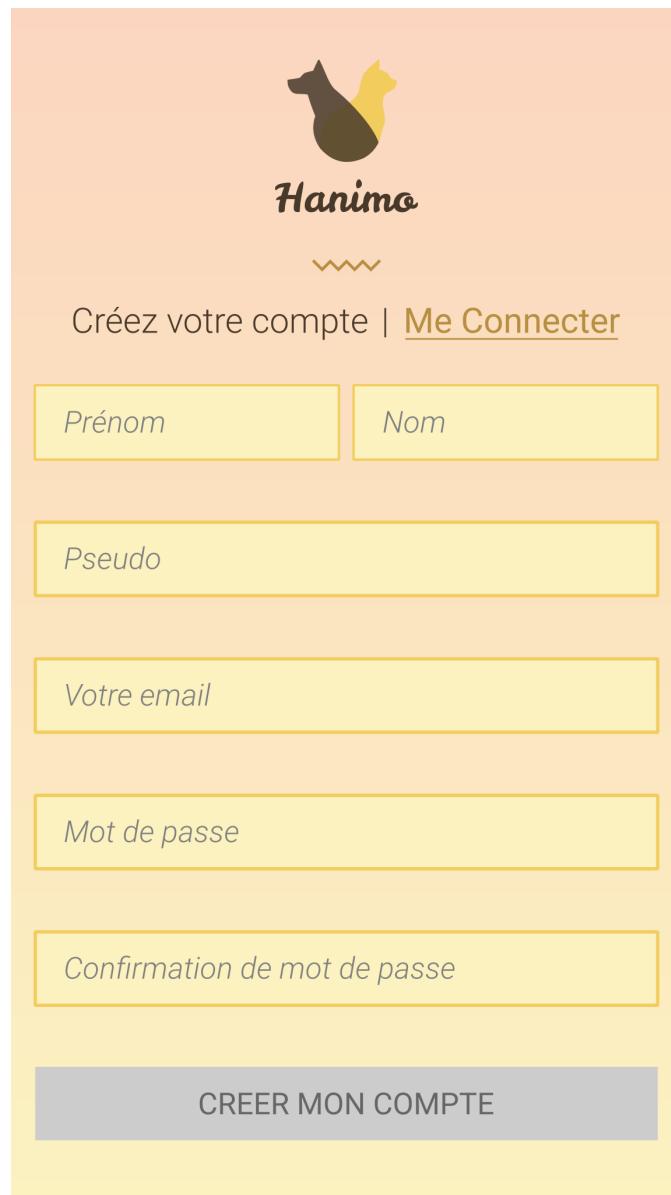


FIGURE 4.13 – Interface d'inscription

## Interface de profil de l'utilisateur connecté

Cette interface présente le profil de l'utilisateur courant. A partir de cette interface il peut accéder au paramètre de son profil pour les modifier , changer sa photo de profil en appuyant sur l'icône de la caméra et visualiser ses activités .

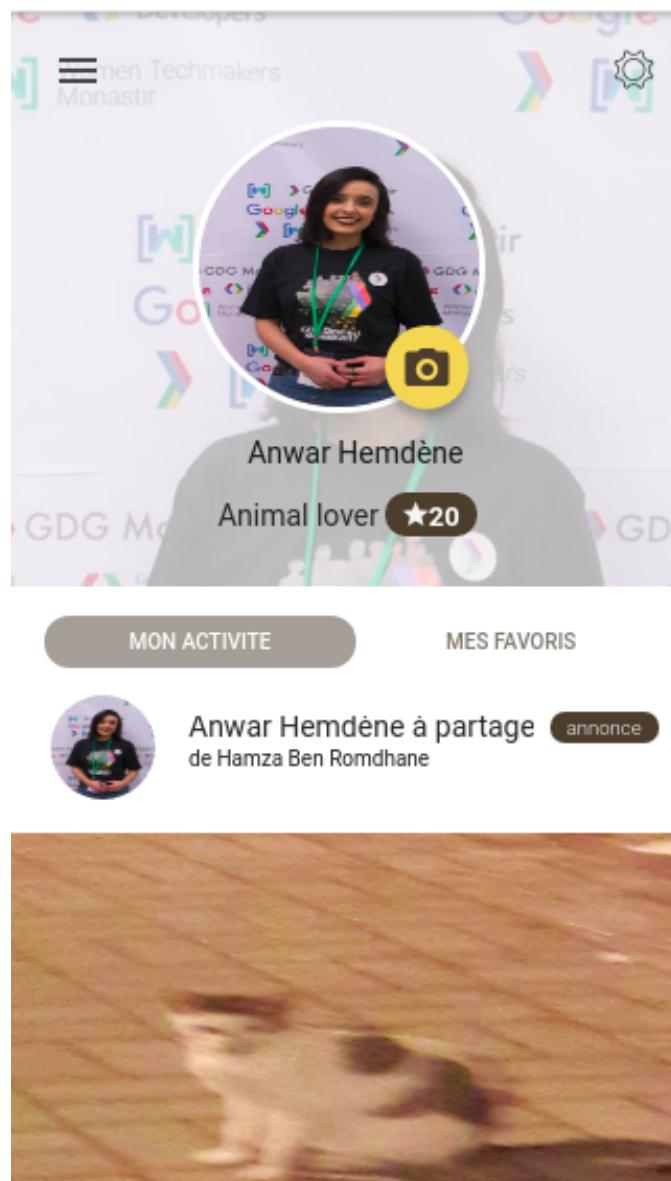


FIGURE 4.14 – Visualisation de profil de l'utilisateur connecté

## Interface de paramètres

Cette interface permet à l'utilisateur de modifier les paramètres de son compte , changer le mot de passe et choisir ses préférences de notifications

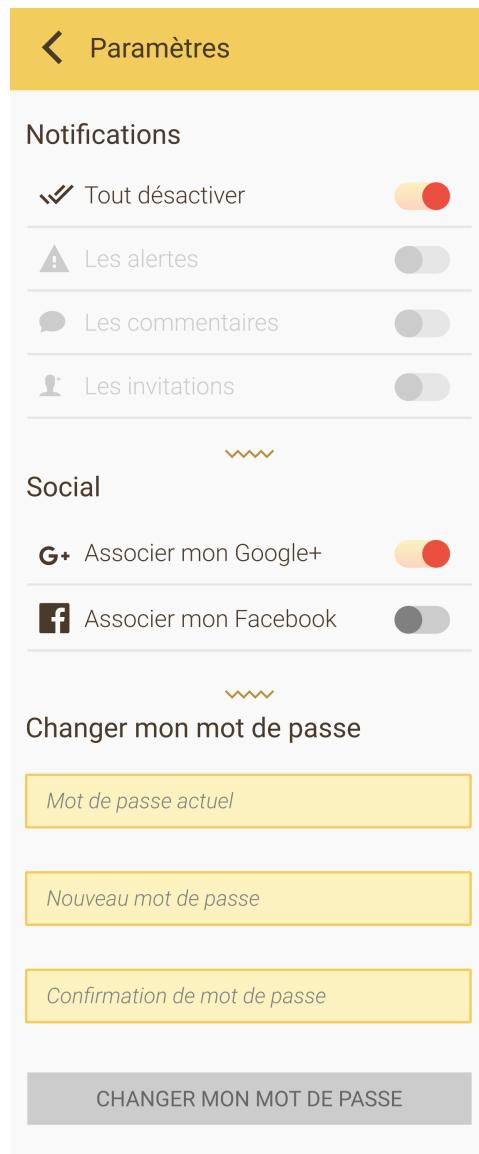


FIGURE 4.15 – Interface des paramètres

### Interface de profil d'un ami(e)

Cette interface présente le profil d'un ami où on peut visualiser ses activités , ses annonces et ses publications . L'utilisateur courant peut supprimer cet ami(e) de sa liste ou lui envoyer un message privé .

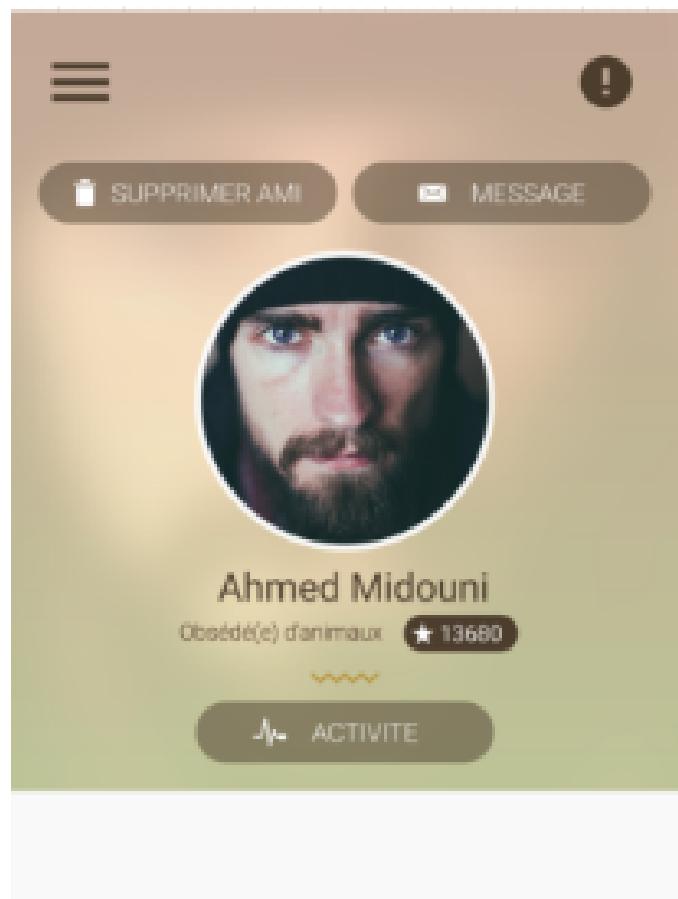


FIGURE 4.16 – Visualisation de profil d'un ami(e)

### Interface d'envoi d'une demande d'ajout en amis

Lorsque l'utilisateur courant accède au profil d'un autre non ami , les activités de ce dernier ne sont pas visible , donc il doit lui envoyer une invitation pour l'ajouter à sa liste d'amis comme le montre la figure suivante .

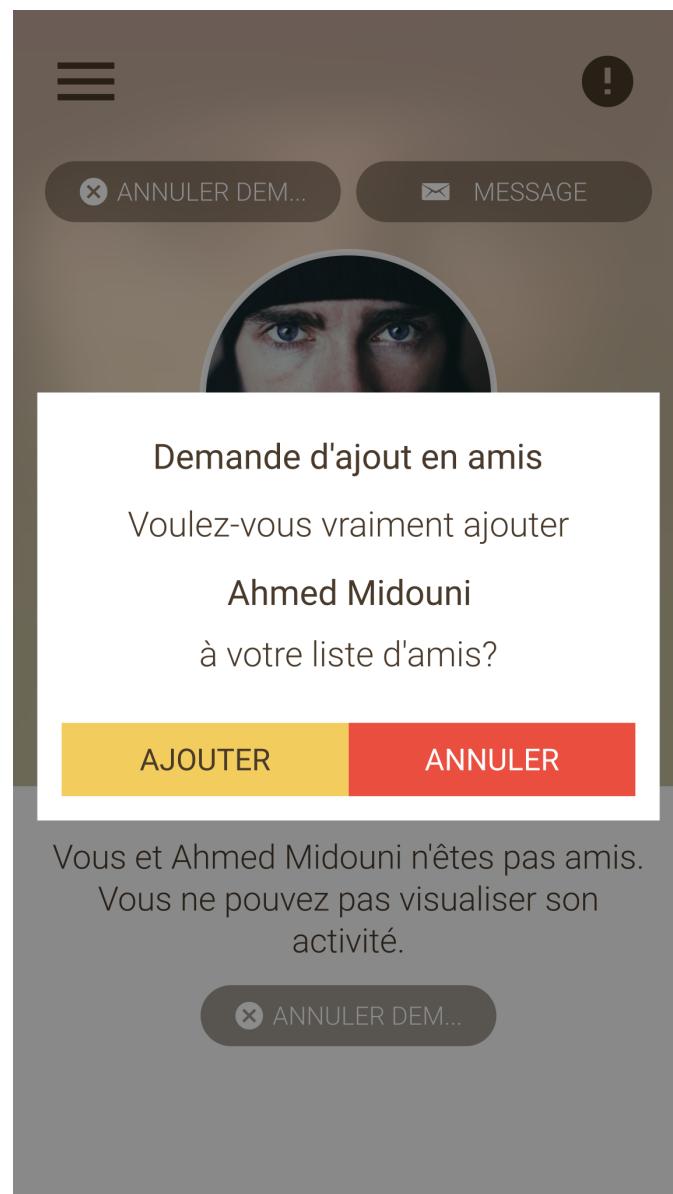


FIGURE 4.17 – Envoyer une demande d'ajout en amis

### Interface d'annulation de l'envoi d'une demande d'ajout en amis

Après l'envoi d'une invitation , l'utilisateur peut changer d'avis et annuler sa demande .

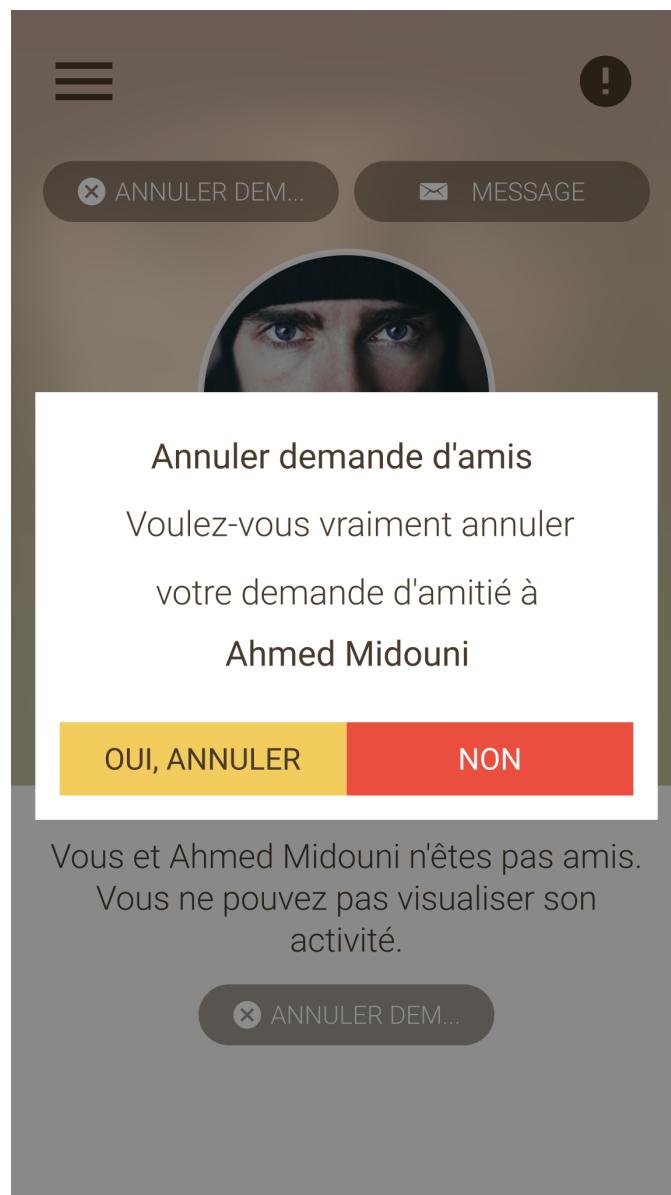
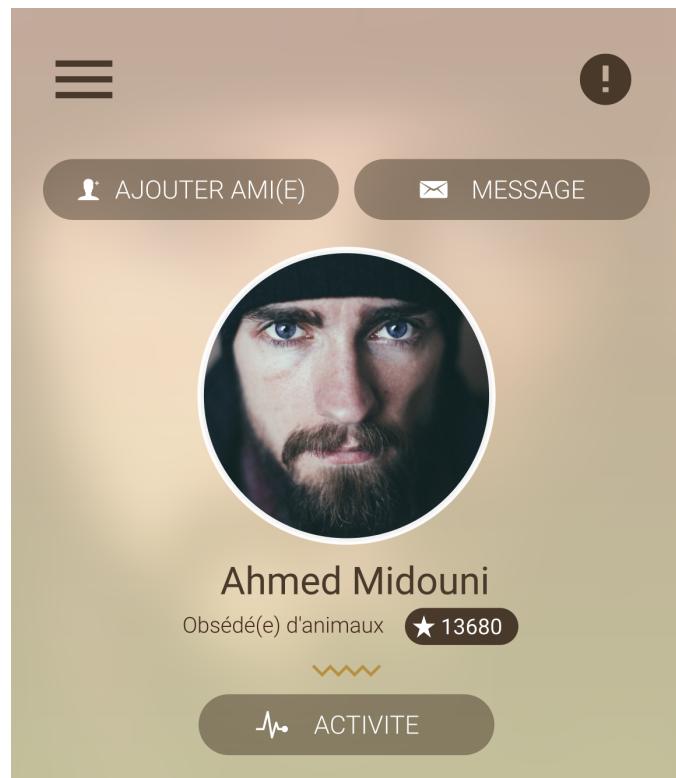


FIGURE 4.18 – Annulation de l'envoi d'une demande d'ajout en amis

### Interface de profil d'un utilisateur non ami

Cette interface présente le profil d'un autre utilisateur non ami(e) avec l'utilisateur courant.



Vous et Ahmed Midouni n'êtes pas amis.

Vous ne pouvez pas visualiser son activité.

[AJOUTER AMI\(E\)](#)

FIGURE 4.19 – Profil d'un utilisateur non ami

### Interface de rédaction d'un message

Cette interface permet de rédiger un message et l'envoyer un ami a partir de l'interface de son profil

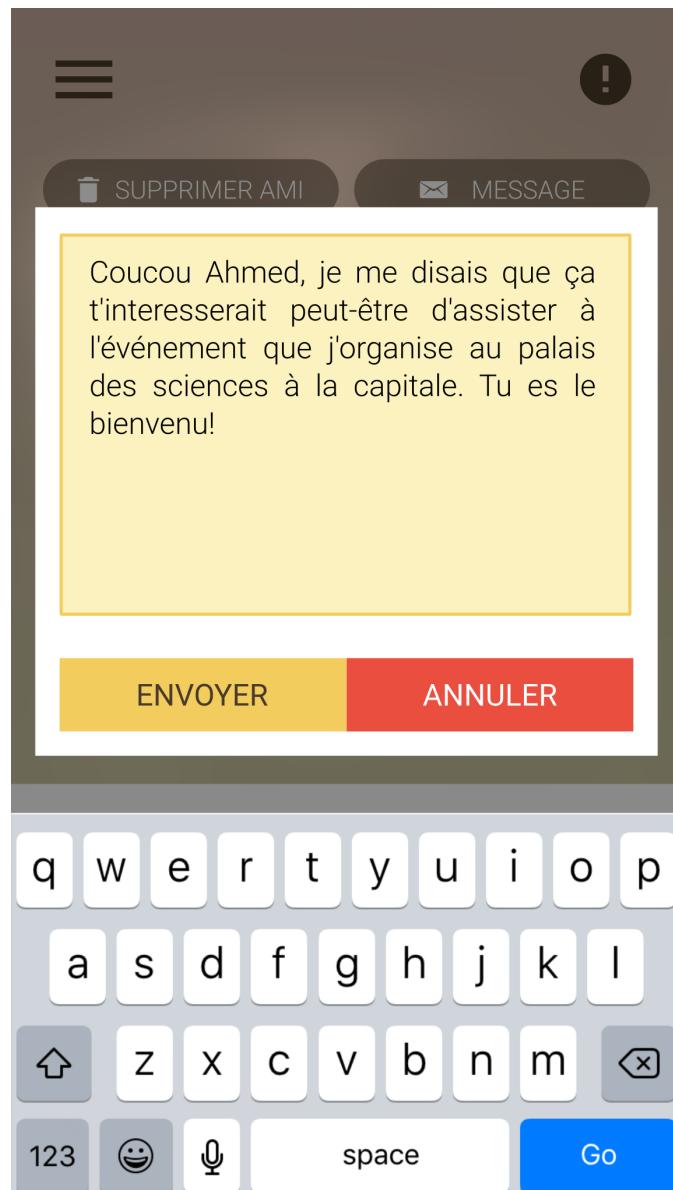


FIGURE 4.20 – Composer un message

## Interface de fil d'accueil

Cette interface présente l'interface d'accueil de l'application où l'utilisateur courant peut visualiser les publications des ses amis et les annonces à proximité. Il peut ainsi commenter , aimer ou partager ces publications .

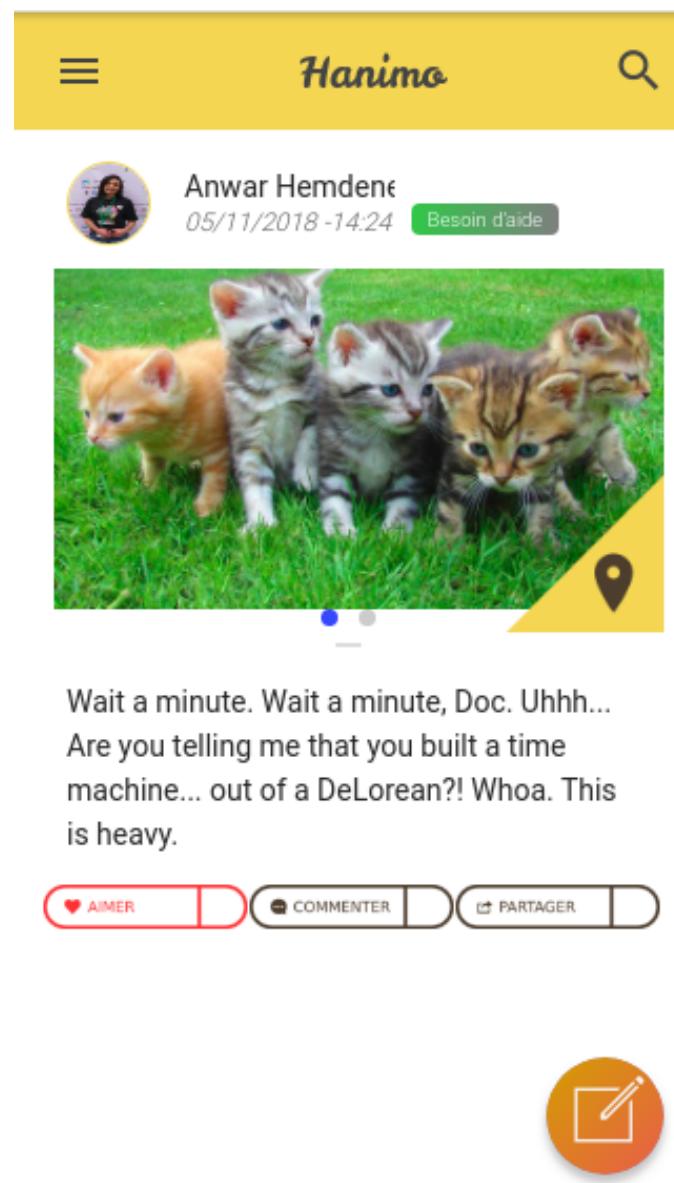


FIGURE 4.21 – Interface de fil d'accueil

## Interface d'ajout d'annonces / publication

A partir de l'interface d'accueil et en appuyant sur le bouton en bas à droit l'utilisateur peut choisir le type de publication/annonce qu'il souhaite publier

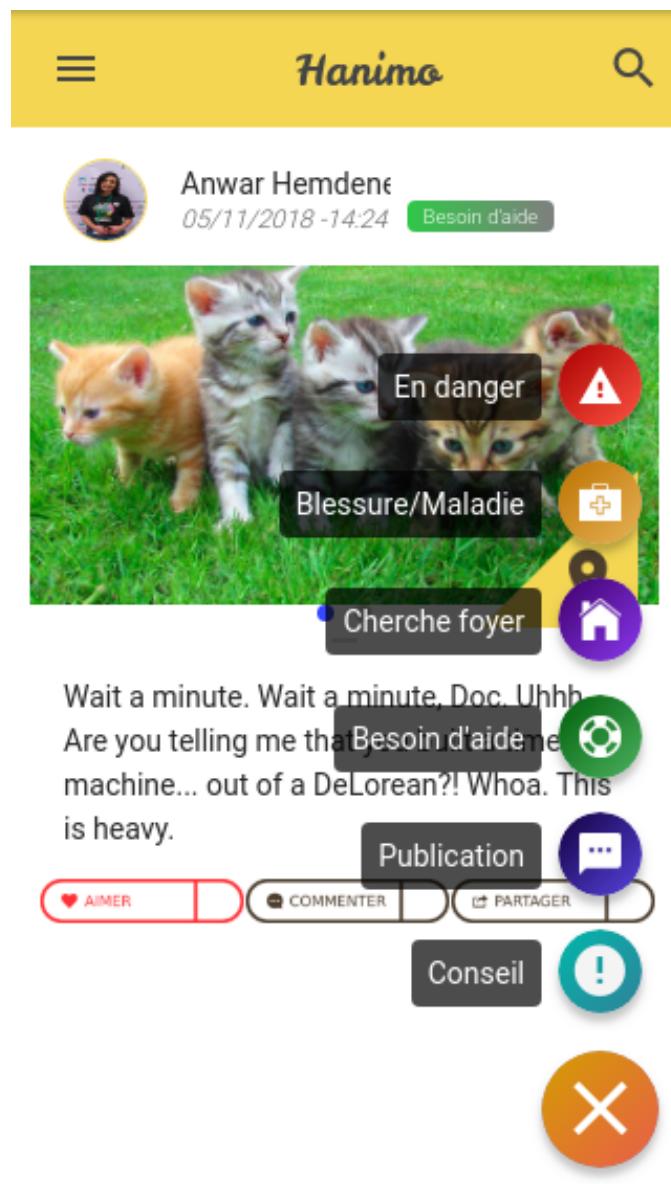


FIGURE 4.22 – Choix de publication de divers types d'annonces

### 4.3.2 Back Office

#### Interface de connexion

Cette interface permet à l'administrateur de se connecter en introduisant son email et son mot de passe .

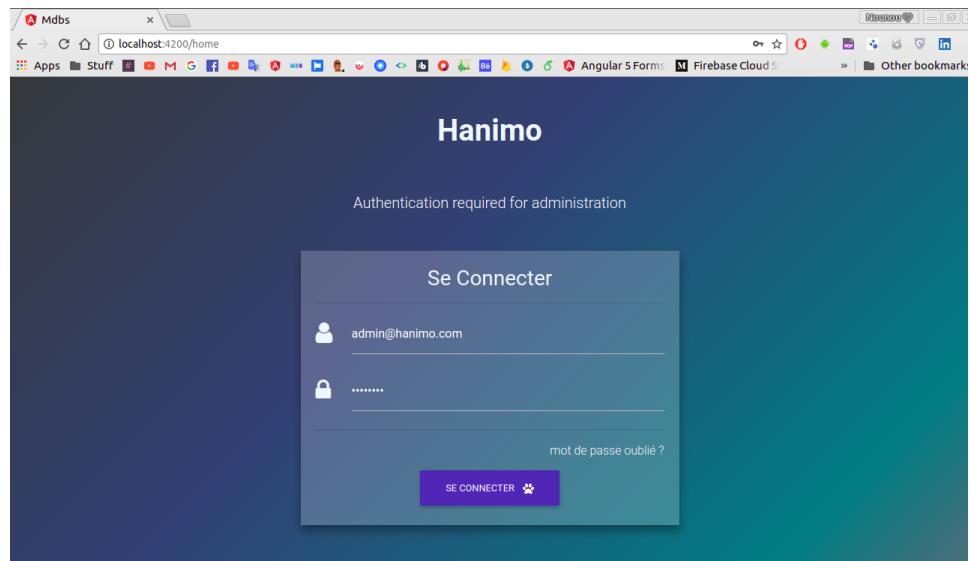


FIGURE 4.23 – Interface de connexion

#### Interface de changement de mot de passe

A partir de cette interface , l'administrateur peut mettre à jour son mot de passe

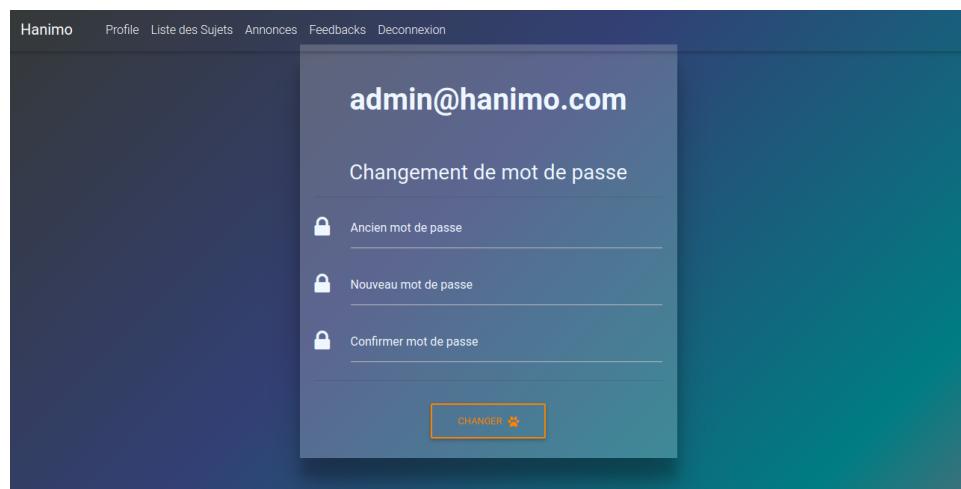


FIGURE 4.24 – Interface de gestion de profil

## Interface d'ajout d'un nouveau sujet

Cette interface permet à l'administrateur de publier un nouveau sujet de discussion en remplissant les champs vides et il peut ajouter une image .

The screenshot shows a web browser window with the URL `localhost:4200/nouveau-sujet`. The page title is "Sujet de discussion". A central modal dialog is titled "Nouveau Sujet". It contains three input fields: "Titre" (Title), "Description", and "Upload Photo". Below the photo upload field is a "Choose File" button with the text "No file chosen". At the bottom of the modal are two buttons: "AJOUTER" (Add) in blue and "RESET" in teal.

FIGURE 4.25 – Interface pour ajouter un nouveau sujet

## Interface des listes des sujets publiés

A partir de cette interface , l'administrateur peut visualiser les sujet publié précédemment et les commenter .

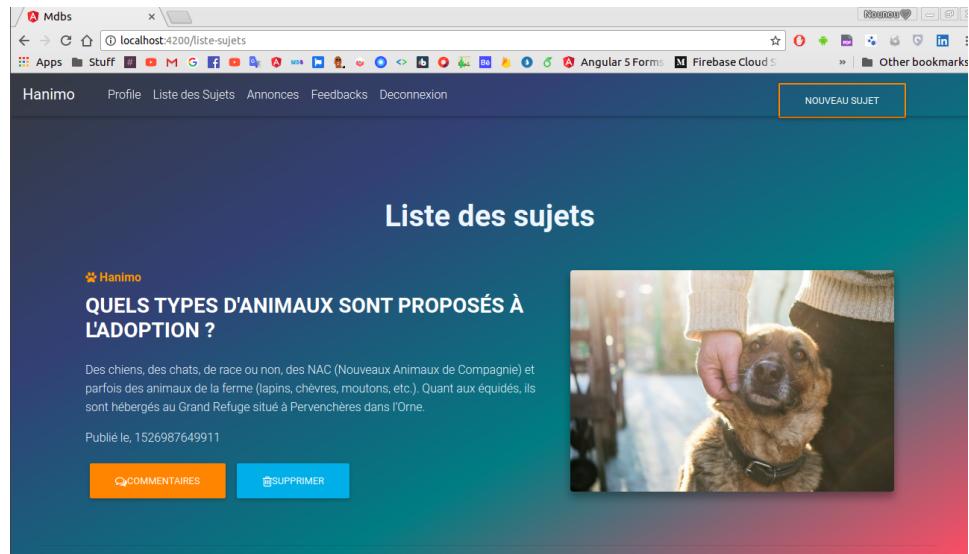


FIGURE 4.26 – Interface des listes des sujets

## Interface des feedbacks

Cette interface présente les feedback envoyé par les utilisateurs où l'administrateur peut répondre à ces feedback .

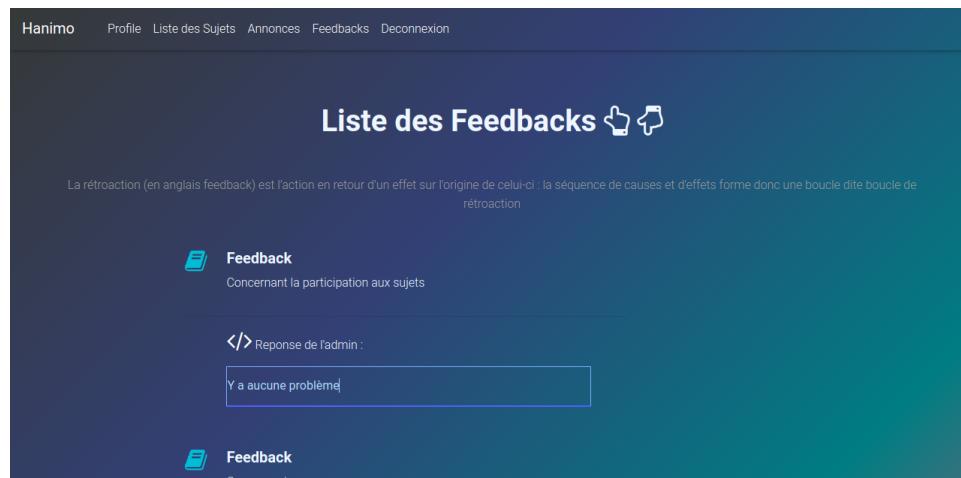


FIGURE 4.27 – Interface des feedbacks

## 4.4 Conclusion

Dans ce chapitre nous avons présenté l'environnement logiciel utilisé lors du projet, suivi par des captures d'écran montrant les interfaces en cours de fonctionnement de notre application mobile et site web.

# Conclusion générale

Ce travail s'inscrit dans le cadre du projet de fin d'études en vue de l'obtention du diplôme de la Licence appliquée en informatique : Systèmes Informatiques et Logiciels. Il a pour objectif de développer une plateforme permettant de venir en aide à des animaux en cas de danger.

Nous résumons ainsi le travail effectué au cours de notre projet. Nous avons mis en place un réseau social dédié aux activistes animaliers. Ce réseau va leur permettre de poster des annonces à propos d'animaux en besoins d'aide ou intervention, partager des expériences en publiant des conseils, échanger des messages privés, personnaliser leurs profils, former un réseau d'amis . Notre application est accompagnée par un back office qui est une partie administrative permettant aux administrateurs d'effectuer des tâches de contrôles tel que la modification de l'état d'urgence d'une annonce, répondre aux feedback et partager des sujets de discussion .

Ce projet était une occasion pour apprendre de nouvelles technologies avancées. Comment réagir dans les situations de blocage et la pression de temps , respecter la spécification et surtout mieux gérer notre temps .

Nous estimons que nous avons pu atteindre la plupart des objectifs fixés. Cependant quelque fonctionnalités reste manquantes comme l'appel vidéo et audio ainsi qu'un manque au niveau du design vu la limite du temps et la taille du projet .

# Bibliographie

- [1] Definition 2TUP :  
<http://www.catchu.fr/accelerateur/la-demarche-projet-2tup/>
- [2] Avantages et inconvénient développement android :  
<https://franceyou.wordpress.com/2013/01/05/avantages-et-inconvenients-de-developpement-d-applications-android>
- [3] Definition UML  
<https://www.math-info.univ-paris5.fr/bouzy/Doc/UML-NotesCours.pdf>
- [4] Architecture 2 tiers image  
<https://www.linkedin.com/pulse/develop-2-tier-web-applications-firebase-marian-veteanu>
- [5] Definition diagramme de classe  
<https://www.supinfo.com/articles/single/3224-uml-diagramme-classe>
- [6] MVC  
[https://www.tutorialspoint.com/mvc\\_framework/mvc\\_framework\\_introduction.htm](https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm)
- [7] Vue de Kruchten  
<https://docs.google.com/document/d/1PoB8jQ216G6iZHtcaku5cb5HFhPSJmPdlFnRtqVnbeA/edit>
- [8] Github - présentation  
<http://simplonline.co/13-ressources/11-github-presentation>
- [9] Inkscape  
<https://inkscape.org/en/>
- [10] Introduction à LaTeX  
<https://www.plpeeters.com/blog/fr/post/110-introduction-a-latex>
- [11] Environnement de développement  
<https://www.blog-nouvelles-technologies.fr/3461/quest-ce-quun-environnement-de-developpement/>
- [12] Firebase  
<https://openclassrooms.com/courses/creez-un-backend-scalable-et-performant-sur-firebase/titre-de-votre-premier-chapitre-222>

- [13] Draw.io  
<https://www.tice-education.fr/index.php/tous-les-articles-er-ressources/articles-internet/819-draw-io-un-outil-pour-dessiner-des-diagrammes-en-ligne>
- [14] Angular  
<http://creersonsiteweb.net/page-angular-js-angularjs-apprendre-javascript-cours-tuto-exemple-example>
- [15] Ionic  
<https://www.supinfo.com/articles/single/155-presentation-framework-ionic>
- [16] Nodejs  
<https://makina-corpus.com/blog/metier/2014/introduction-a-nodejs>
- [17] GANTT  
<http://www.gantt.com/fr/>