



Plan de conception

Version 1



Groupe 2 :

Fatima-ezzahra Mendada

Nouhaila Khaouti

Table des matières

I.	Introduction :.....	2
II.	Les règles de gestion :.....	2
III.	Diagramme de classe :	3
	↳ Description textuelle :	3
IV.	Diagramme de cas d'utilisation :	4
	↳ Définition des diagrammes :	4
	❖ Description des cas d'utilisation	5
V.	Diagramme de flux :.....	7
VI.	Résumé :.....	9

I.Introduction :

La conception est une phase importante avant la réalisation de tout projet. Cette phase joue un rôle primordial dans le progrès rapide de la partie codage. Elle nécessite des méthodes permettant de mettre en place un modèle sur lequel on va s'appuyer c'est-à-dire créer une représentation similaire à la réalité, de telle façon à faire ressortir les points auxquels on s'intéresse. Pour ce travail, j'ai opté pour le langage UML.

UML est une boîte à outils offrant des éléments de modélisation adaptés à l'approche objet. De plus, elle permet de décrire les différents aspects de notre application par un large panel de diagrammes.

II.Les règles de gestion :

- ✚ Règle 1 : user, notification -> one to many.
- ✚ Règle 2 : user, commentaire -> one to many.
- ✚ Règle 3 : user, donation ->one to many.
- ✚ Règle 4 : user, demande-> one to many.
- ✚ Règle 5 : user, checklist-> one to many.
- ✚ Règle 6 : user, sitting->one to one.
- ✚ Règle 7 : user, post-> one to one.
- ✚ Règle 8 : post, animal-> one to one.
- ✚ Règle 9 : post, notification->one to many.
- ✚ Règle 10 : post, user-> one to one.
- ✚ Règle 11 : notification, post->one to one.
- ✚ Règle 12 : notification, user -> one to one.
- ✚ Règle 13 : animal, post-> one to many.
- ✚ Règle 14 : animal, category->one to one.
- ✚ Règle 15 : category, breed ->one to many.
- ✚ Règle 16 : category, animal -> one to many.
- ✚ Règle 17 : category, sitting->one to many.
- ✚ Règle 18 : sitting, category-> one to many.
- ✚ Règle 19 : sitting, type-> one to many.
- ✚ Règle 20 : sitting, demande ->one to many.
- ✚ Règle 21 : sitting, user-> one to one.

III. Diagramme de classe :

Le diagramme de classe est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation.

Ce diagramme montre la structure interne, il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser le cas d'utilisation.

👉 Description textuelle :

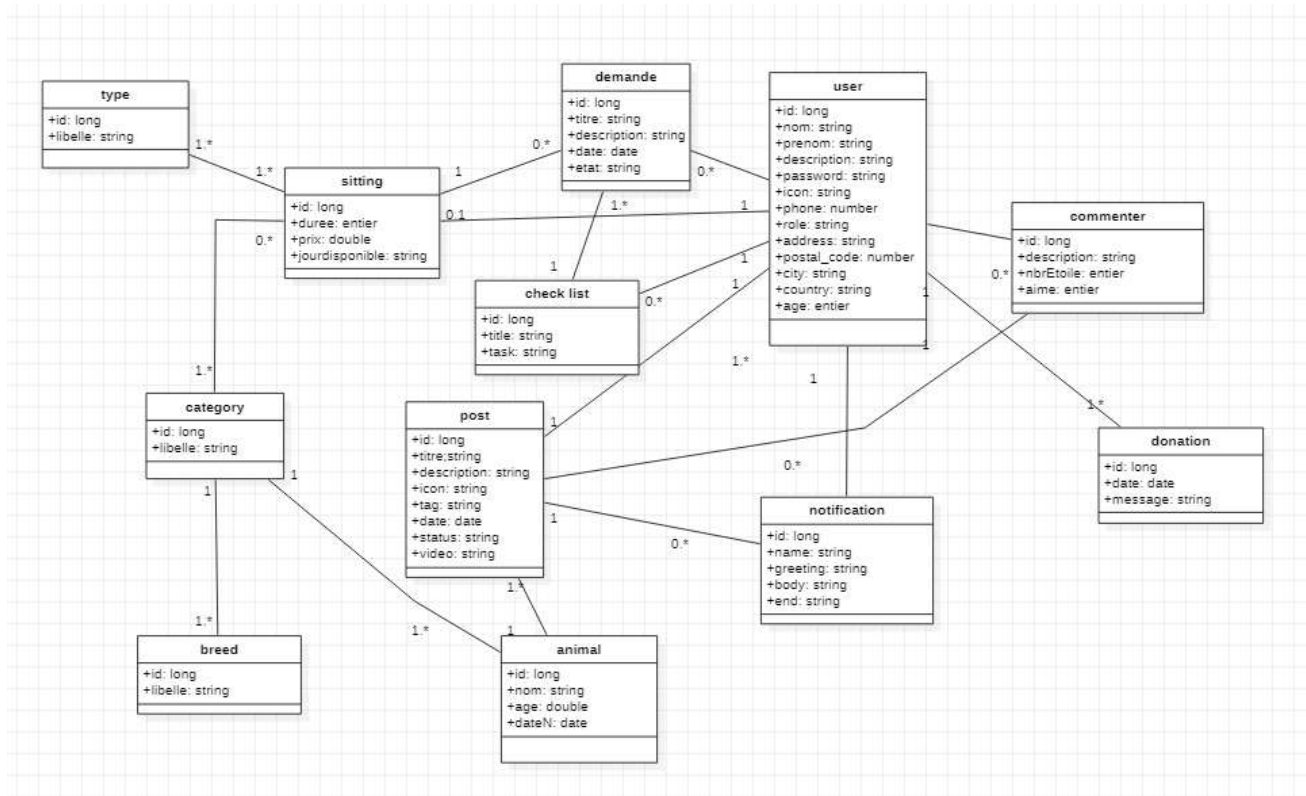


Figure 1: diagramme de classe

Dans le cadre de mon application, la classe Utilisateur représente la classe principale dans ce projet. Un utilisateur possède les attributs suivants : un login, un mot de passe, une description, une icône, un nom et un prénom. Qui a une relation avec plusieurs autres classes qui sont :

- **Contact** : tele1, tele2, un email.
- **Adresse** : adresse, rue, code postal, ville.
- **Demande** : titre, description, date, état.
- **Post** : titre, description, icône, tag, date, statuts, vidéo. -> relation avec animal (nom, Age, date de naissance). -> relation avec catégorie (libelle). -> relation avec breed(libelle), sitting), notification.
- **Donation** : date, message.
- **Commenter** : description, nombre d'étoiles, j'aime.
- **Notification**: nom, greeting, body, end. -> relation avec post.

- Chick List : titre, task. -> relation avec demande.
- Sitting : durée, prix, jours disponibles. : -> relation avec type (libelle), catégorie(libelle), demande.

IV. Diagramme de cas d'utilisation :

Dans la partie précédente, on a présenté le diagramme de classe qui décrit la structure interne du système, or ceci est insuffisant, c'est pour cela que on va traiter les diagrammes de cas d'utilisation, et de séquence qui décrivent le comportement dynamique entre les acteurs et les objets de système.

↳ Définition des diagrammes :

Diagramme de cas d'utilisation :

Le diagramme de cas d'utilisation décrit les utilisations requises d'un système, ou ce qu'un système est supposé faire. Les principaux concepts de ce diagramme sont les acteurs, cas d'utilisation.

❖ Identification des acteurs :

Un acteur est la représentation d'un rôle joué par des entités externes (utilisateurs, dispositifs, matériels ou autre système) qui interagit directement avec le système.

D'après le cahier de charge, on a pu définir les trois acteurs suivants :

Administrateur :

Il se charge de gérer les comptes des étudiants et des utilisateurs, capable d'effectuer toutes les opérations sur l'utilisateur/blog/user.

Utilisateur normale :

Il a beaucoup des fonctionnalités : faire la recherche de sitter ou blog ou pet, créer un profil normal, faire une donation, créer un post de donation et d'autres fonctionnalités qu'on a va citer dans le diagramme de cas d'utilisation.

Utilisateur gardienne :

Il a comme fonctionnalités la consultation des demande offres par l'utilisateur normale, et répondre aux ces propositions.

❖ Description des cas d'utilisation

Après identification des acteurs et après une analyse réflexive, voici les différentes fonctionnalités gérées par l'administrateur, les utilisateurs normaux et les utilisateurs gardienne de mon application sous forme de diagramme de cas d'utilisation.

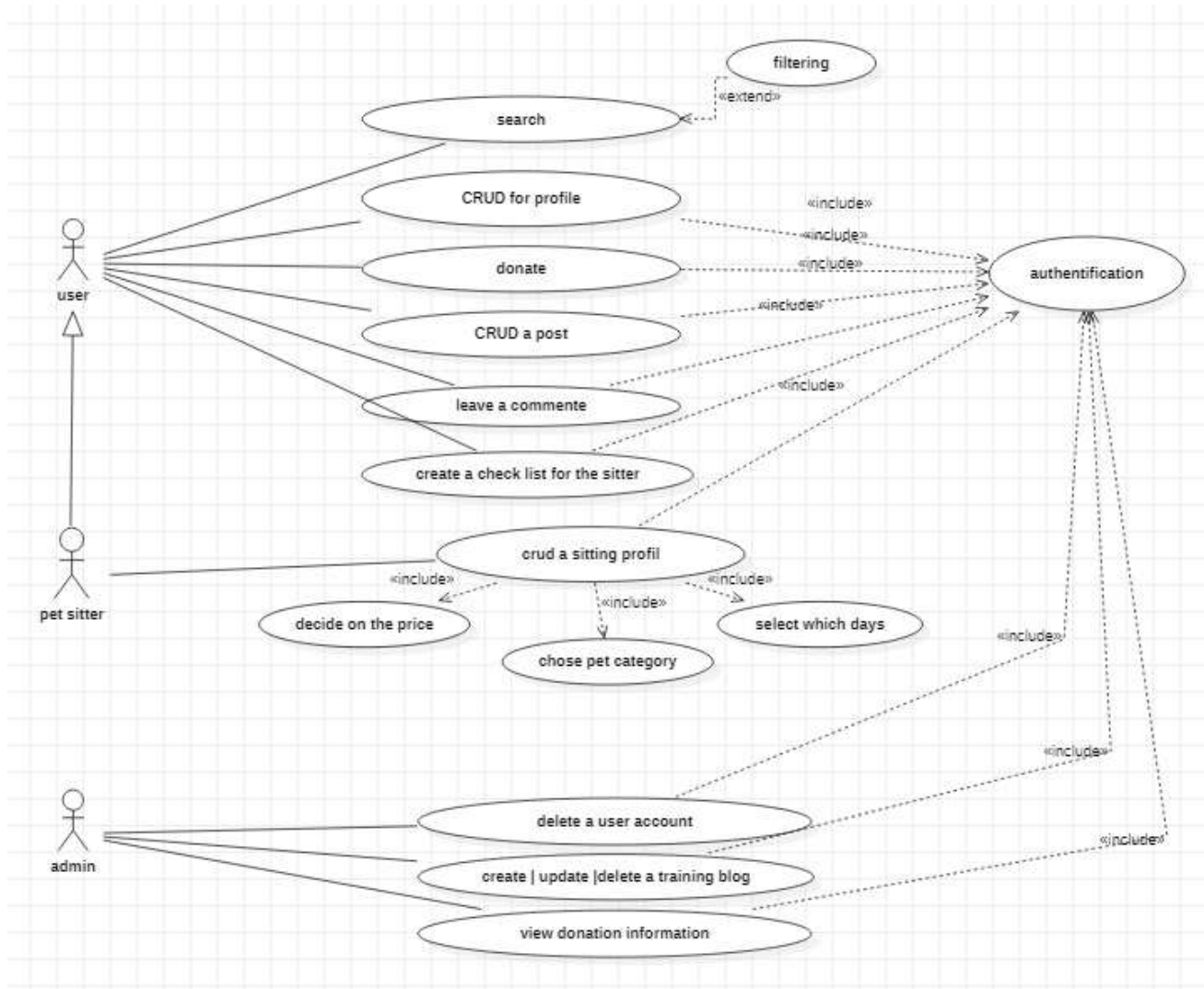


Figure 2 : Diagramme de cas d'utilisation

Cas d'utilisation	Description
Chercher	Pour un blog Pour un sitter Pour un animal d'adoption
Gérer le profil	Delete un profil Update un profil Create un profil
Donation	Donation de l'argent Donation de la nourriture
Gérer un post	Delete un article Update un article Create un article
Checklist	Delete une liste des tâches Update une liste des tâches Create une liste des tâches
Commentaire	Delete un commentaire Update un commentaire Create un commentaire

V. Diagramme de flux :

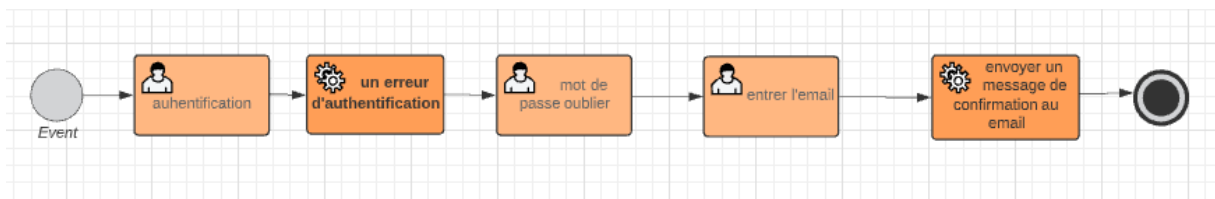


Figure 3 diagramme de processus d'authentification

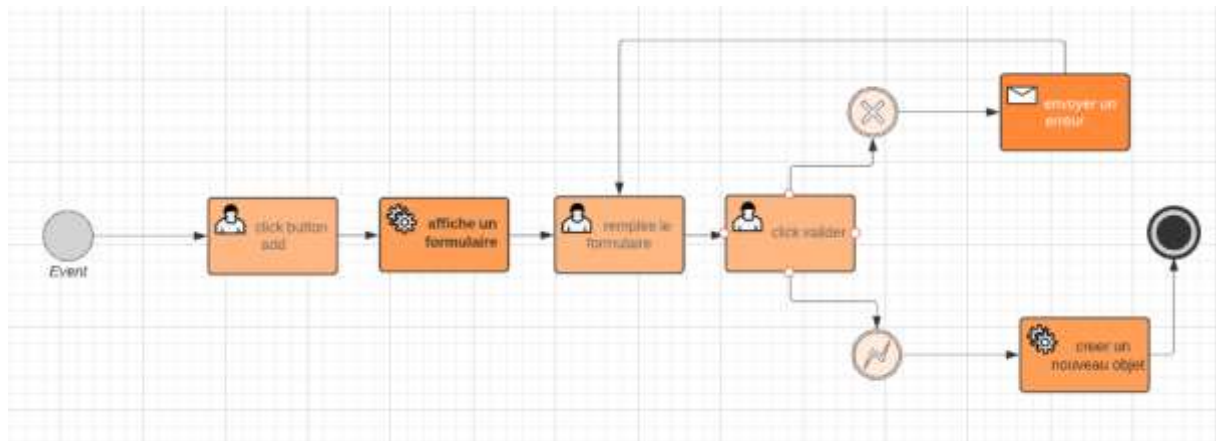


Figure 4 diagramme de processus d'ajout

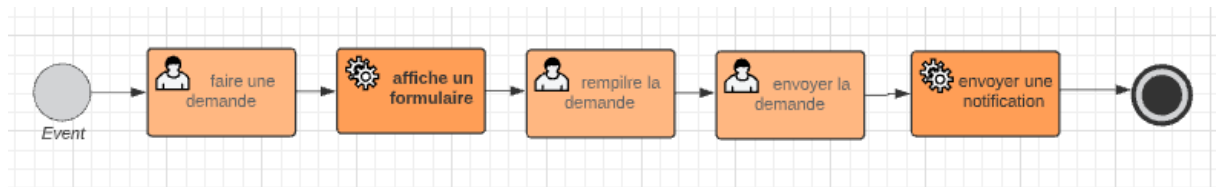


Figure 5 diagramme de processus de demande

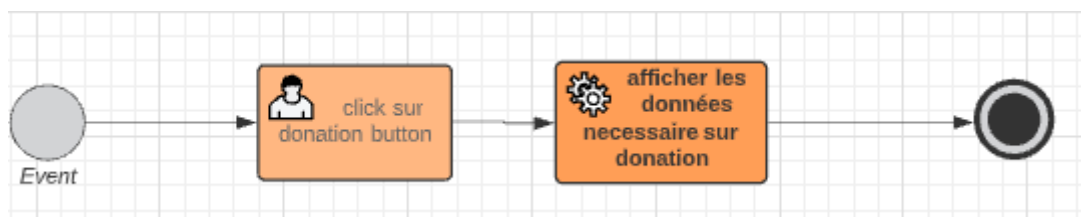


Figure 6 diagramme de processus de donation

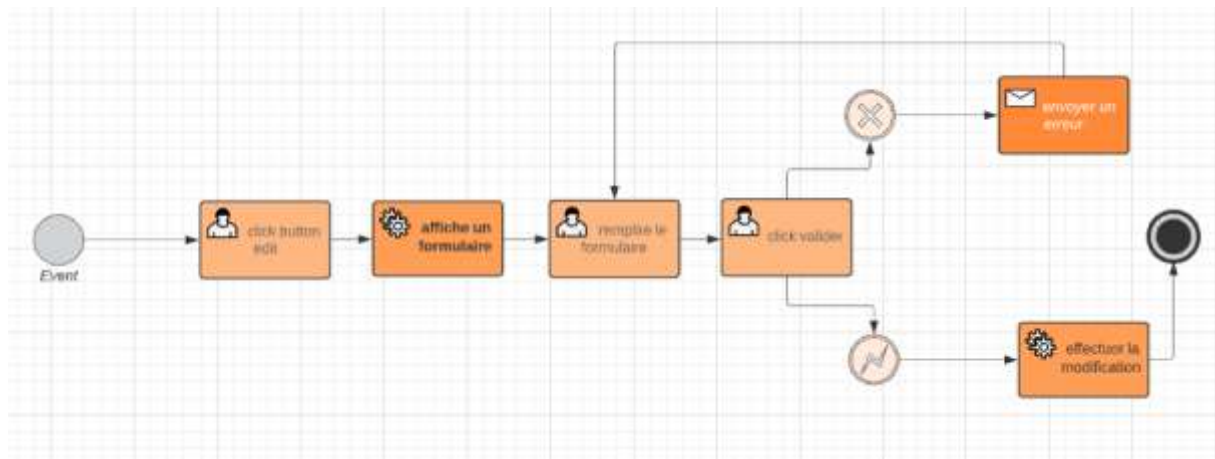


Figure 7 diagramme de processus d'édition

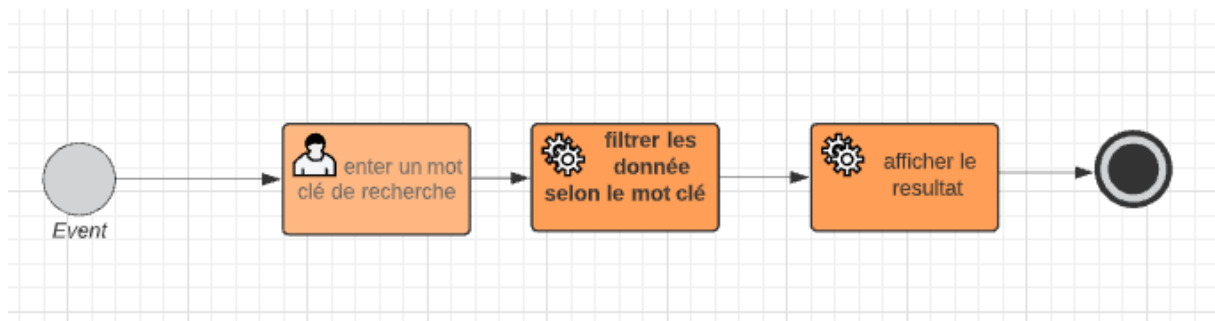


Figure 8 diagramme de processus de recherche

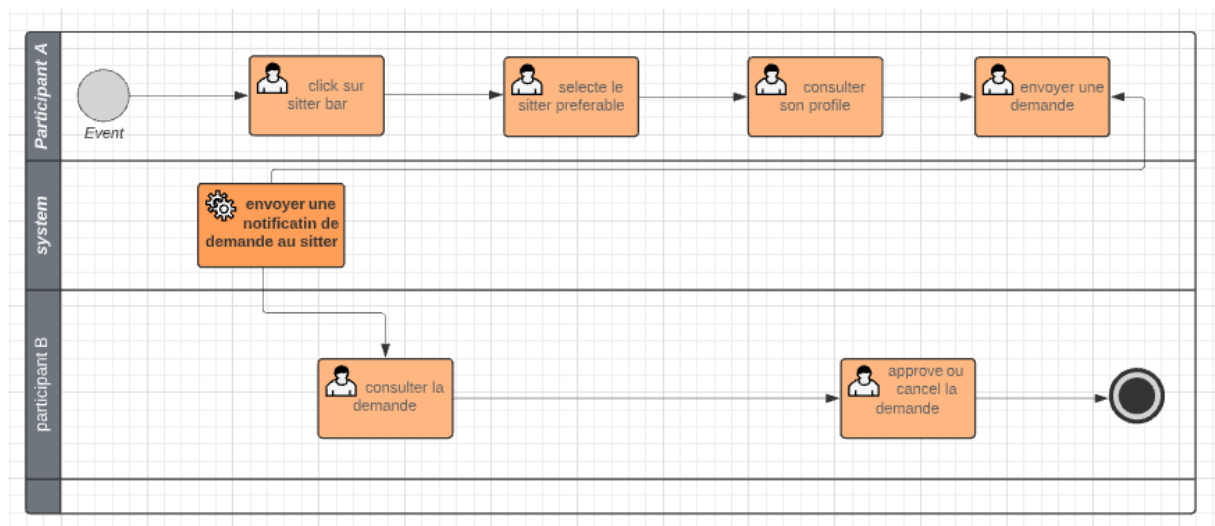


Figure 9 diagramme de processus entre user et sitter

VI. Résumé :

Dans ce chapitre, on a décrit nos diagrammes de cas d'utilisation et de classe, ainsi que les scénarios possibles afin de préparer un terrain favorable pour développer notre application. Dans le chapitre suivant, on va présenter les différents outils et technologies nécessaires pour le développement de notre système.