

CyClic

DOCUMENTATION TECHNIQUE



DUSCHENE Florian, LANDAUER Mathieu, LEDDA Damien,
MOUGENOT Antonin, SCHAEFFER Léonard

Table des matières

I.	Contexte	3
I.a.	Qu'est-ce que CyClic ?	3
I.b.	Objectifs du document technique	3
II.	Concepts	4
II.a.	Thématique.....	4
II.b.	Nom et logos	4
II.c.	Charte graphique	5
II.c.1.	Couleurs	5
II.c.2.	Polices de caractères	5
II.d.	Maquettes.....	6
III.	Analyse.....	10
III.a.	Acteurs.....	10
III.b.	Diagrammes	11
III.b.1.	Classes	11
III.b.2.	Cas d'utilisation.....	12
III.b.3.	Etats-transitions.....	13
III.b.4.	Séquence	16
III.c.	Base de données	18
III.b.1.	Dictionnaire des données.....	18
III.b.2.	Modèle conceptuel de données	19
III.b.3.	Modèle logique de données	20
III.b.4.	Script de création de la base.....	20

Historique de validation

Date	Rédacteur	Version	Commentaire
2021.12.01	Ledda Damien	0.001a	Tentative de doc technique pour des idées de charte graphique du projet
2022.03.02	Les Symfonistes Croustillants	0.1	Remplissage et finalisation de la doc technique

I. Contexte

I.a. Qu'est-ce que CyClic ?

CyClic est un site qui a pour but de donner une seconde vie aux objets inutilement accumulés et/ou délaissés en proposant aux utilisateurs de réaliser des dons. Pour cela, ces derniers peuvent publier une annonce afin de trouver un potentiel receveur et se débarrasser de l'objet concerné en proposant un rendez-vous avec cet autre utilisateur afin de lui transmettre celui-ci. La prise de contact est facilitée entre les différentes parties en donnant la possibilité d'indiquer la localisation dans laquelle la recherche de l'objet est effectuée et grâce à une messagerie personnalisée.

I.b. Objectifs du document technique

Ce document a pour but de détailler le procédé de création du site, de sa genèse à sa version finale.

II. Concepts

II.a. Thématique

CyClic se démarque des autres projets de la formation par le fait qu'il n'a pas de client précis : tout le monde est client. Et il est parti de suggestions réalisées au sein du groupe. Ces suggestions étaient, entre autres :

- Un site permettant de répertorier des cartes collectées,
- Un glossaire de jeux divers,
- Un site thématique ayant pour focus principal les *kaiju*,
- Un jeu thématique à la *Hearthstone Battlegrounds*,
- Un site de location de médias (livres, DVD...),
- Un site de réservation pour un restaurant ou une pizzeria.

Finalement, l'idée d'un site de location a été privilégié, non sans changer son objectif principal : il ne s'agira plus de louer, mais de donner ; et les objets à donner ne sont plus spécifiques.

II.b. Nom et Logos

Pour cela, il fallait déjà avoir un nom et un logo, vu que ce projet sera créé *from scratch*. Vu que son nom de projet, *CroustiRecycle* – un portemanteau entre « Croustillant » et « Recycler », ne faisait pas vraiment professionnel, l'idée de le renommer *CyClic* a été adoptée, notamment du fait que l'objectif du site est de permettre aux utilisateurs de pouvoir donner une seconde vie à leurs objets inutilisés.

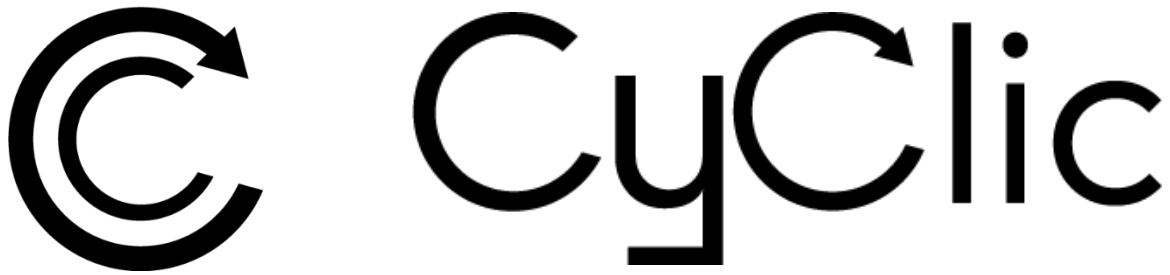
Tandis que le nom est assez rapidement adopté, le logo, quant à lui... a dû passer par des prototypes.



Les prototypes partageaient d'une même idée : remplacer un « C » par une flèche se bouclant, afin de donner l'idée principale dès la vue de son logo. Le concept de vouloir avoir deux polices de caractères différentes est gardée dans la version finale, mais il avait fallu faire divers tests pour avoir

une idée de quelles polices allaient ensemble. Entre la Klavika et la Brown (ci-dessus), ou l'Ariel avec une police script (à droite), on a vu qu'aucune police qu'on avait ne convenait. Du moins, jusqu'à ce que...





Ce logo, réalisé en plein cours d'UI-UX, réalise parfaitement ce que nous avons voulu avoir lors des tests : un message clair (offrir une seconde vie aux objets) et une cohésion entre les caractères. De plus, il est possible de le décliner en quelques couleurs :



II.c. Charte graphique

II.c.1. Couleurs

Les couleurs utilisées sont variées, mais restent dans un même pattern :

Charte principale : #edb20c #18edb1 #1844ed #ee7c01 #ed3f0c
 Charte secondaire : #001894 #00bf01

Note :

Même si la charte principale est utilisée sur le site, les documentations comme celle-ci peuvent utiliser la secondaire, notamment si elles ne sont pas orientée vers l'utilisateur, mais plus vers les développeurs.

II.c.2. Polices de caractères

Les polices de caractères utilisées sont choisies assez méticuleusement afin de sortir du lot tout en restant fidèles à un site à but non-lucratif. Sont utilisées :

- **Cy** : Cette police particulière, non pas parce que son nom coïncide avec celui du projet (CyClic), d'où son utilisation sur le fragment « **Cy** » du logo, rentre parfaitement comme police pour headers par le fait qu'elle soit géométriquement imprévisible, comptant au total 802 glyphes.
- **Outfit** : Cette contraste avec la précédente par son côté plus lisible, plus amical, d'où son utilisation dans le corps de texte.

Note :

D'autres polices avaient été prévues à la base, mais la combinaison des deux mentionnées fonctionna le mieux pour ce site.

II.d. Maquettes

Pour le design de CyClic, un benchmarking a eu lieu entre les sites à vocation similaire. Des constats ont été réalisés : un site sobre, à thème clair et avec une charte graphique qui ne se limite qu'à quelques couleurs. Quelques exemples montrent la direction au niveau de l'interface :



← La page d'accueil.
L'approche épurée a été
remarquée dans les
résultats du benchmark, et
l'utilisation de la charte
principale a été privilégiée
car pouvant couvrir les
différences entre les
éléments du site.



Inscription

Nom

Prénom

Adresse

CP Ville

Adresse email

Mot de passe

Confirmer le mot de passe

☐ L'utilisateur doit être connecté au moins 15 minutes avant de pouvoir accéder à l'interface et doit être enregistré. Cliquez ici pour en savoir plus.

S'inscrire

Vous avez déjà un compte? [Se connecter](#)



Inscription

Nom

Prénom

Adresse

CP Ville

Adresse email

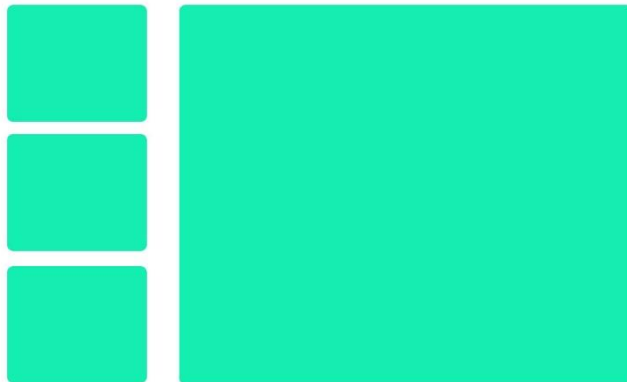
Mot de passe

Confirmer le mot de passe

S'inscrire

Vous avez déjà un compte? [Se connecter](#)

^ L'inscription (donc par extension, la connexion) optent pour une approche différente du reste du site. Seul l'essentiel est affiché : le logo permettant de retourner à l'accueil et le formulaire d'authentification. Rien d'autre. La page de création d'une nouvelle annonce se basera sur ces pages, mais inclueront le header et le footer.



Titre annonce

état : bidule

catégorie : bidule

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Quis ipsum suspendisse ultrices gravida. Risus commodo viverra maecenas accumsan lacus vel facilisis.



Pseudonyme

★★★★★ 7 éavis

Contacteur



Afficher dans Google



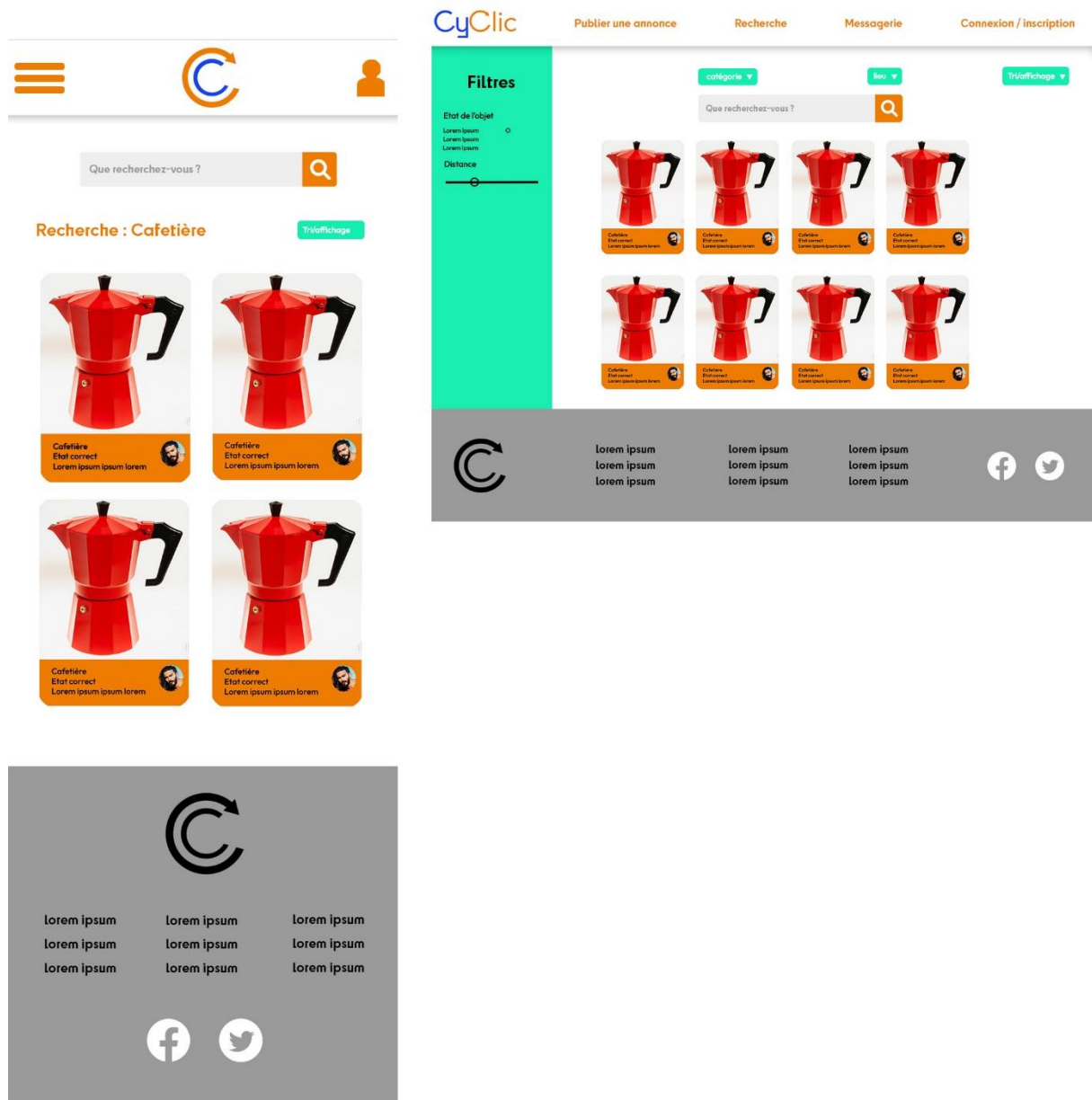
lorem ipsum
lorem ipsum
lorem ipsum

lorem ipsum
lorem ipsum
lorem ipsum

lorem ipsum
lorem ipsum
lorem ipsum



^ La page de l'annonce se résume à l'essentiel, sans superflu : son titre, ses images, son état, sa catégorie, son auteur, sa note et sa localisation. Il est aussi possible de le contacter via un bouton fait pour.



^ La page de recherche. Avec les filtres et les tris.

Note :

La version finale diffère de ces maquettes par le fait que d'autres ajouts ont été réalisés entre temps, et une adaptation a dû être réalisée à cause de contraintes liées au responsive.

III. Analyse

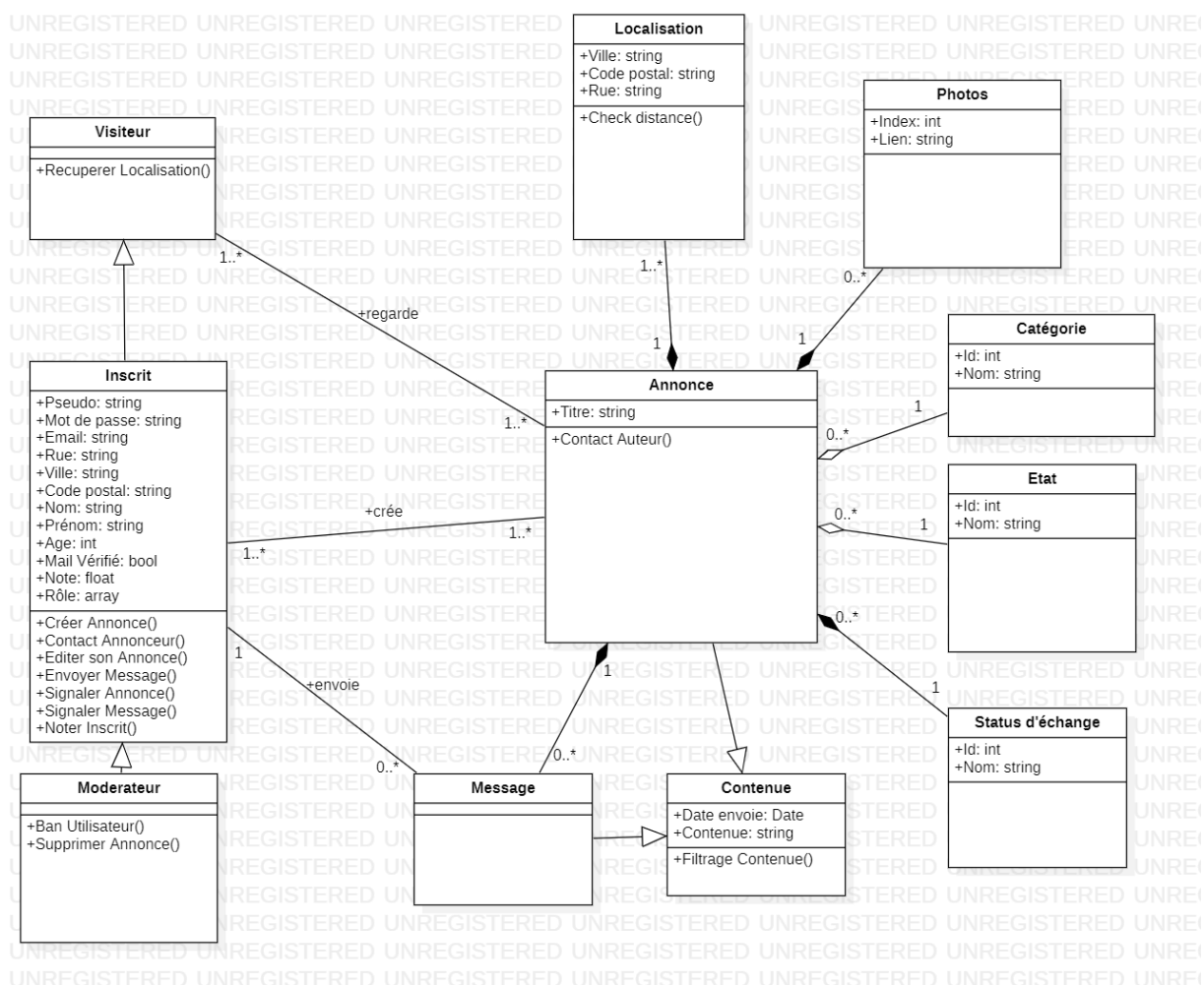
III.a. Acteurs

Chez CyClic, les acteurs sont répartis en 3 catégories :

- Le visiteur, qui peut :
 - o Visiter le site et faire une recherche d'annonces
 - o Consulter une annonce précise qui lui aurait tapé dans l'œil.Toutefois, s'il veut faire plus que ce qui lui est attribué, il devra se connecter.
- L'utilisateur inscrit, héritant des droits du visiteur, qui peut en plus :
 - o Créer une annonce concernant un objet dont il n'a plus besoin
 - o Interagir avec des annonces d'autres utilisateurs
 - o Envoyer des messages et prendre rendez-vous avec un autre utilisateur afin de réaliser un don
 - o Noter un utilisateur après un échange.
- Le modérateur/l'administrateur, héritant des droits d'un utilisateur inscrit, a en plus la possibilité de :
 - o Gérer les catégories (en ajouter, en supprimer, en éditer) et les états
 - o Supprimer les annonces qui ne sont pas conformes aux règles du site
 - o Contrôler les signalements d'utilisateurs (et, par extension, bannir les utilisateurs abusifs)

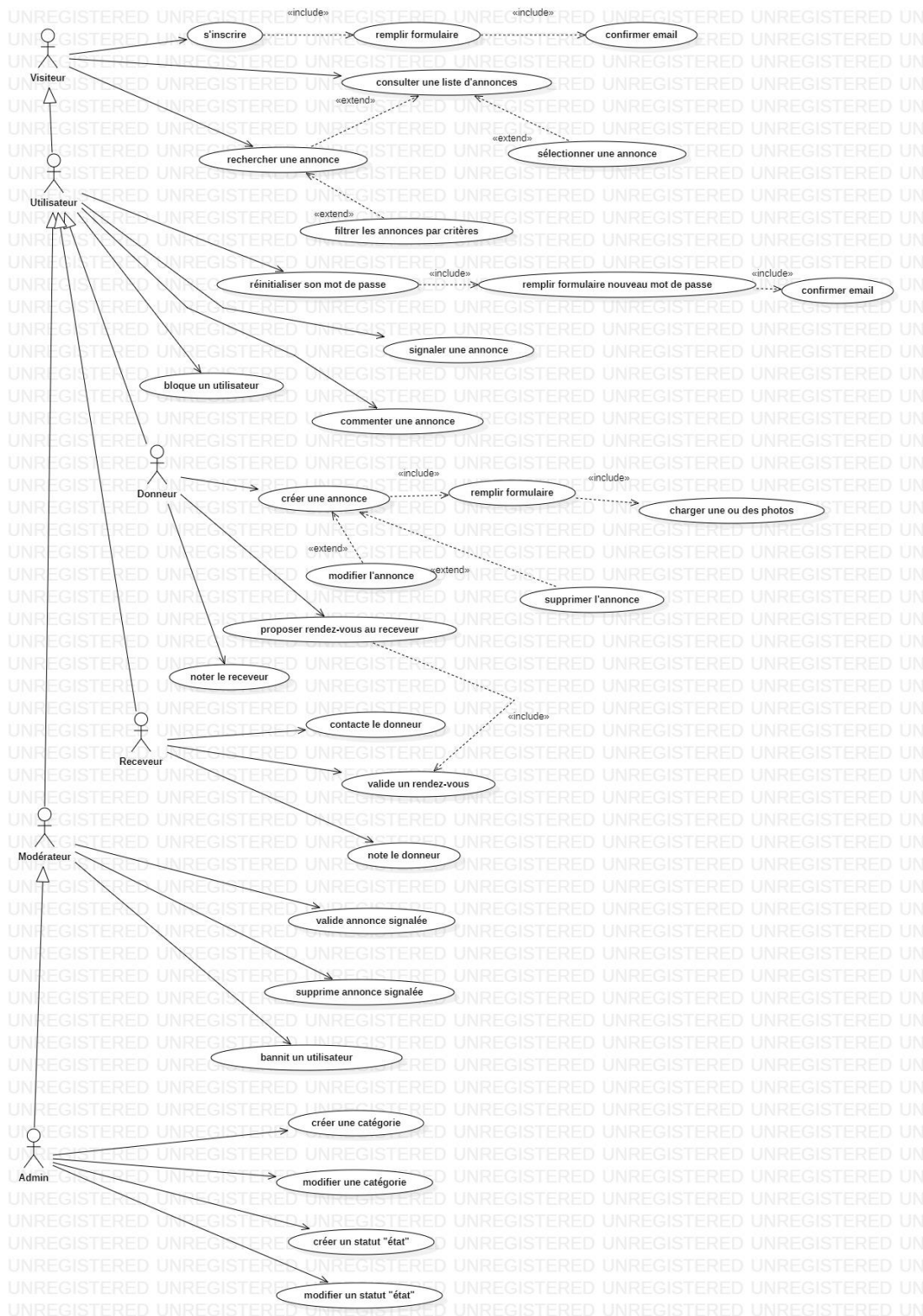
III.b. Diagrammes

III.b.1. Classes



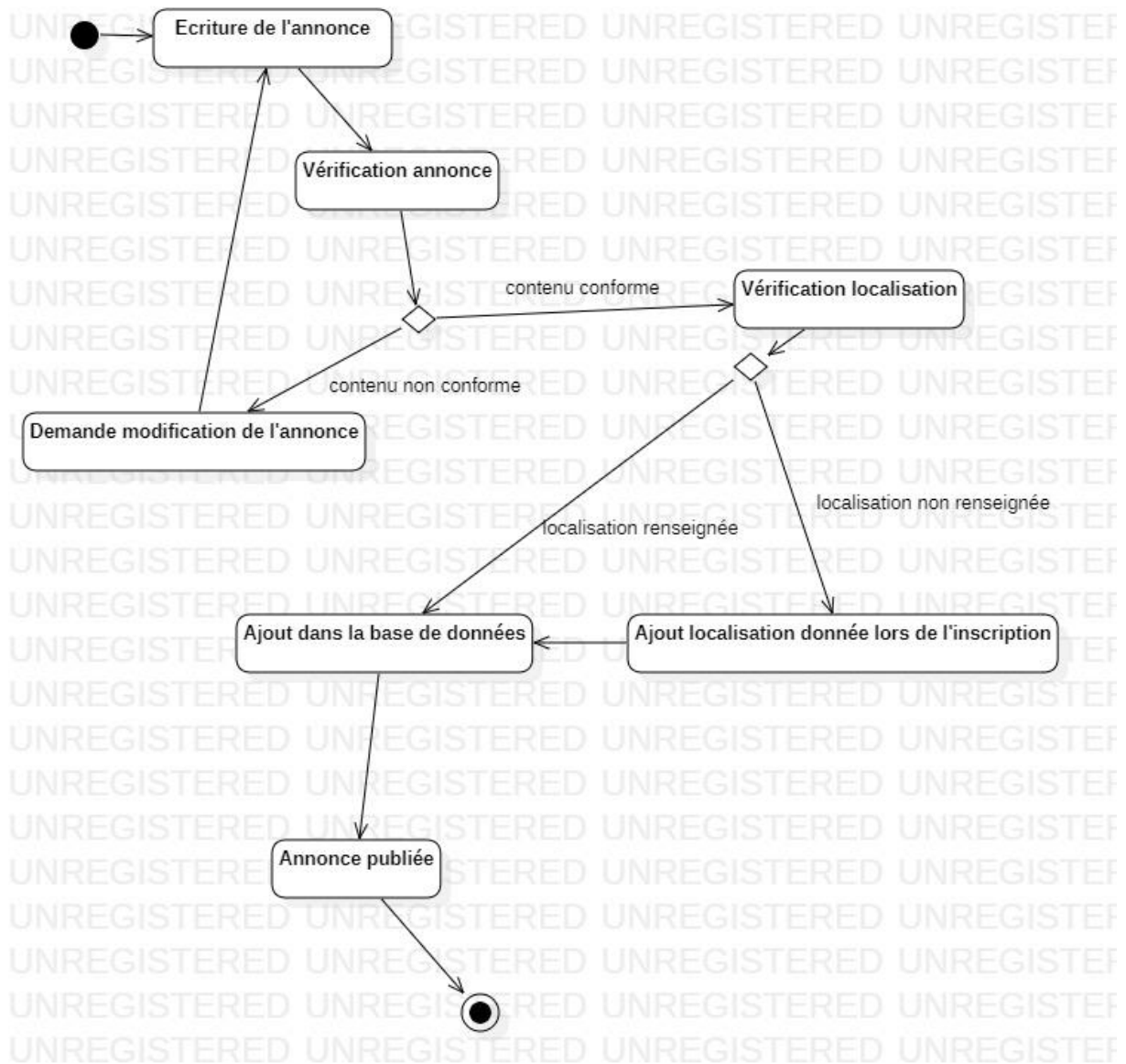
Le diagramme de classe montre les différentes possibilités d'action offertes à l'utilisateur en fonction de son statut. Il présente également les différents attributs liés aux différentes entités ainsi que les liens d'héritage (un modérateur hérite des caractéristiques d'un inscrit) et de possession (une photo appartient à une annonce).

III.b.2. Cas d'utilisation

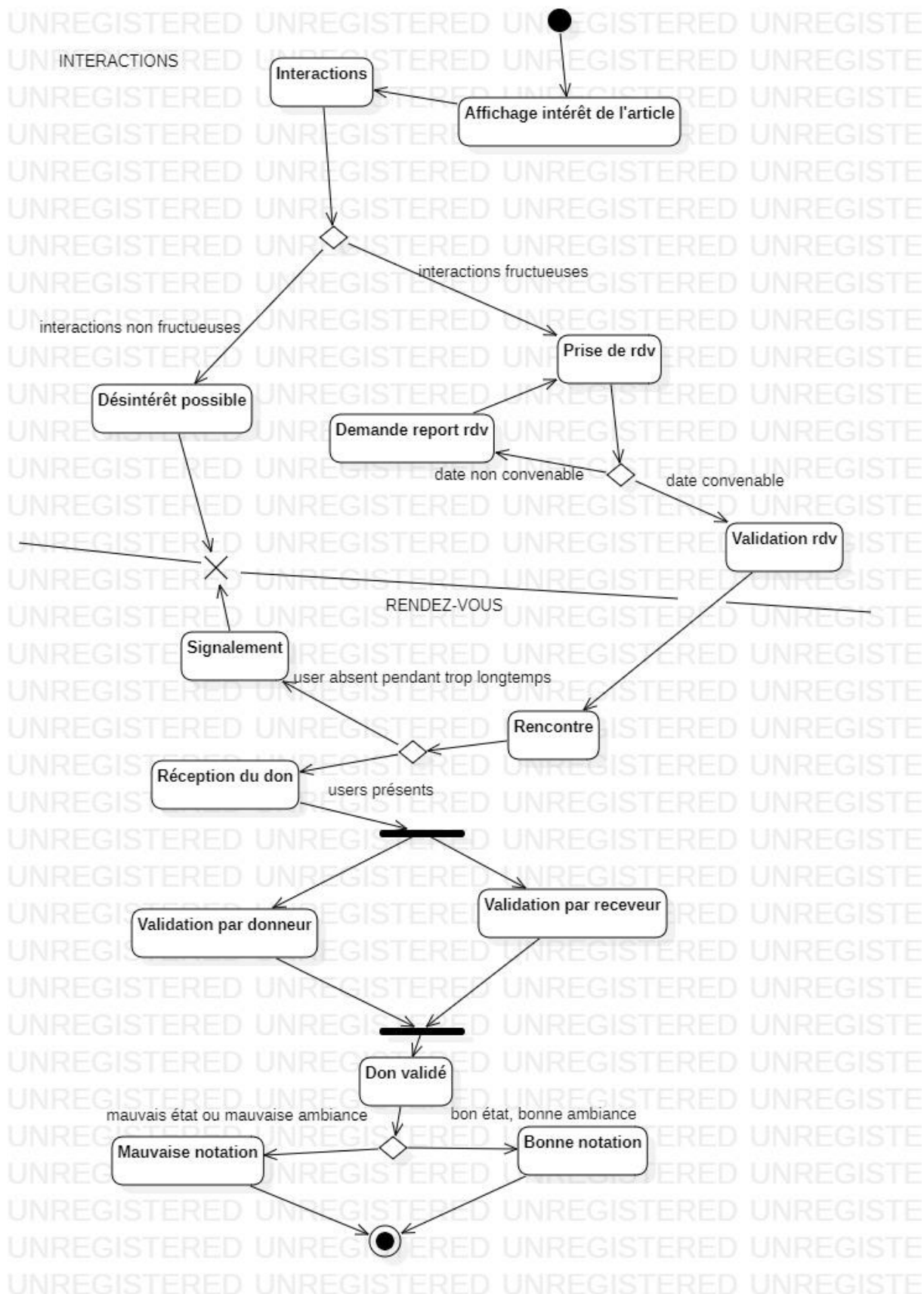


Ce diagramme de cas d'utilisation présente toutes les actions que peut réaliser un utilisateur en fonction de son statut/rôle.

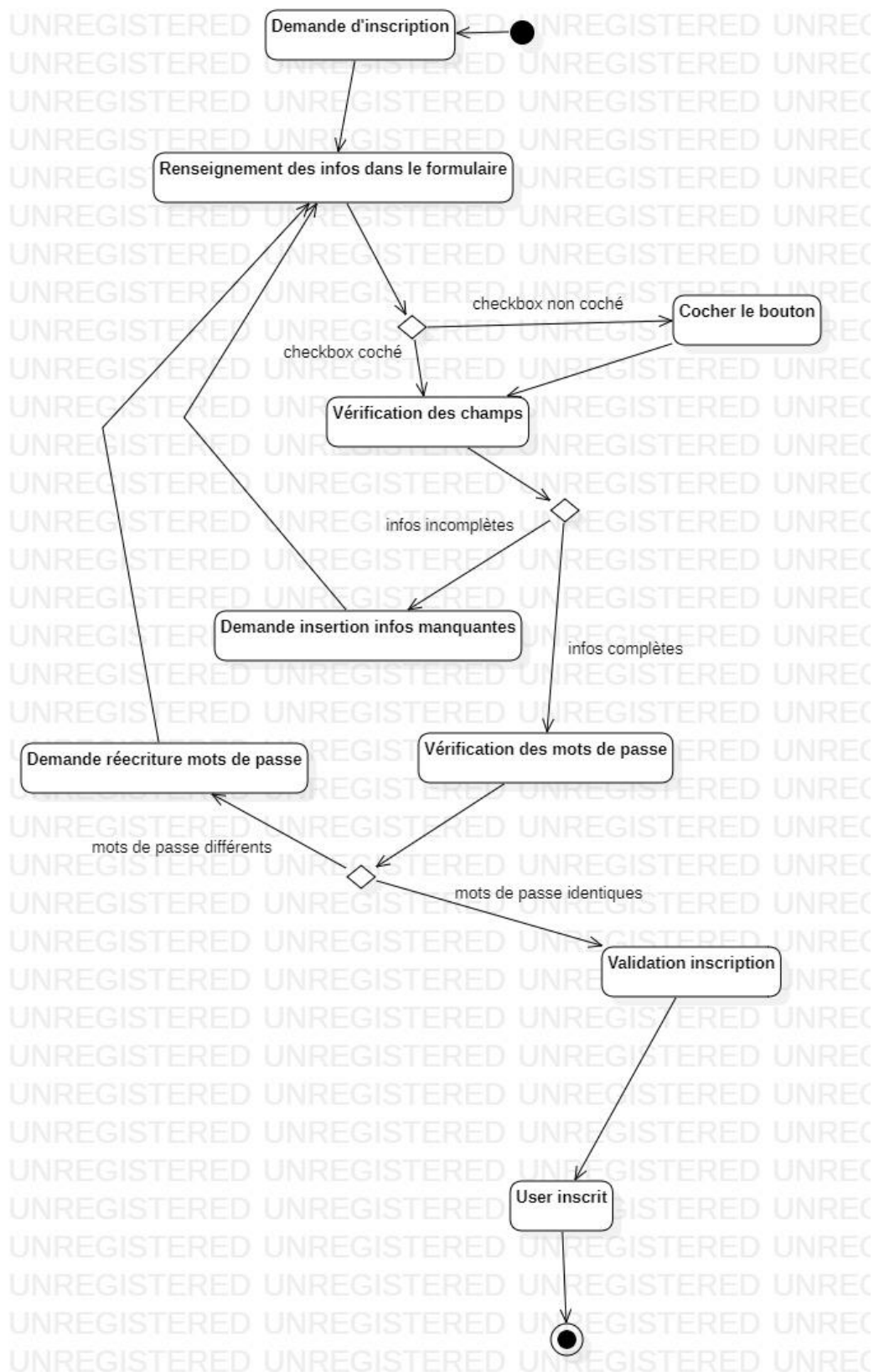
III.b.3. Etats-transitions



Ce diagramme montre les différents états et transitions d'une annonce lors de sa publication.

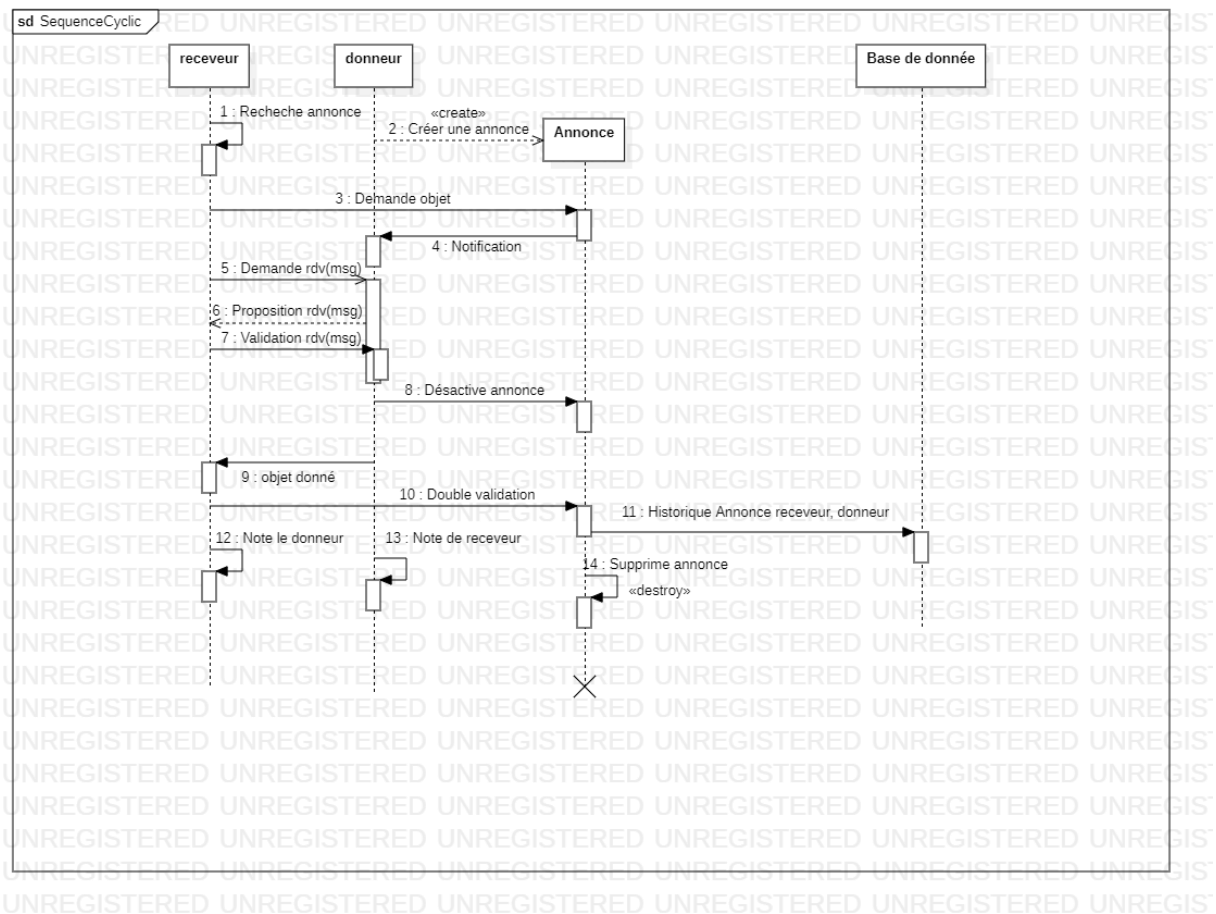


Ce diagramme montre les différents états et transitions de la prise de contact d'un donneur, à la remise de l'objet qui finit par la notation entre utilisateur.

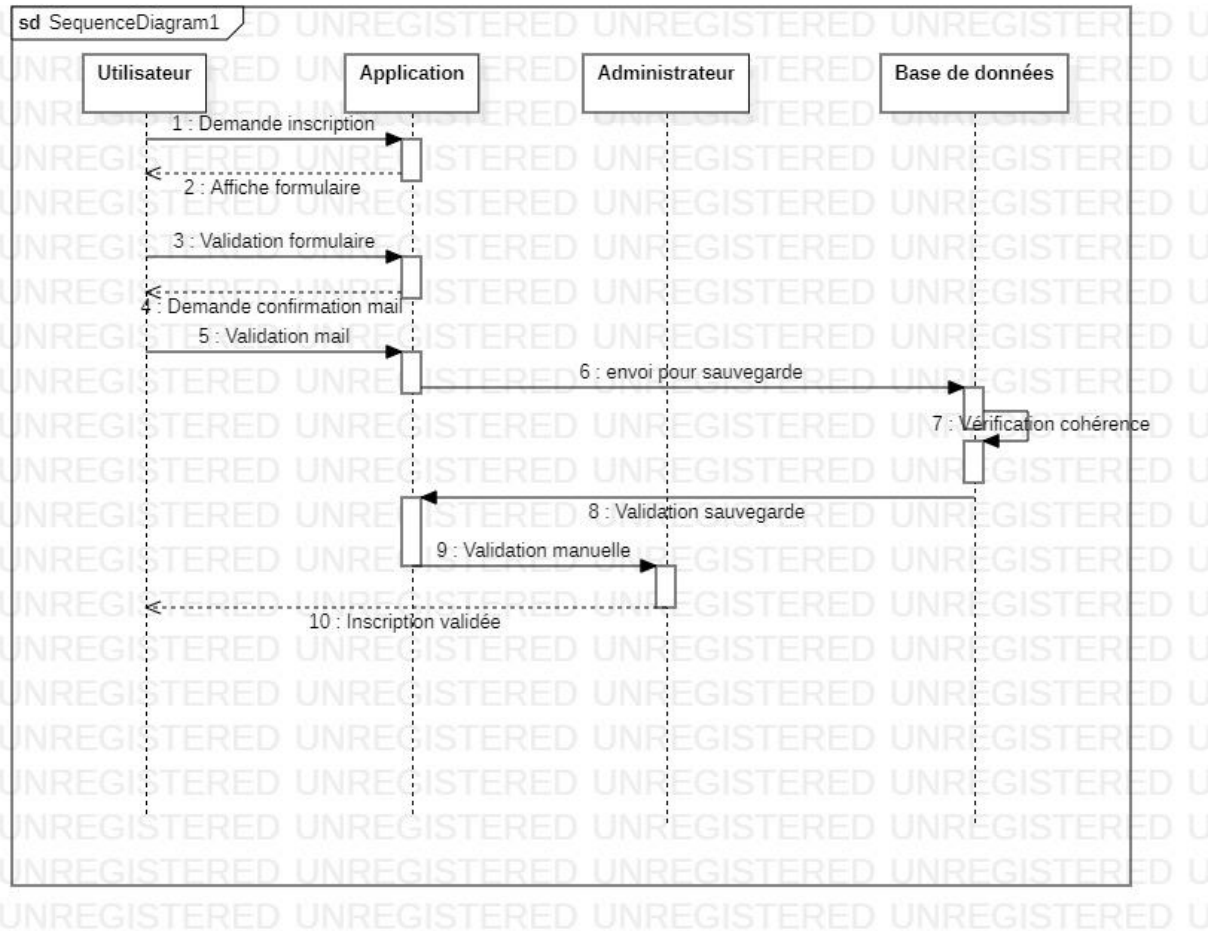


Ce diagramme montre les différents états et transitions pour s'inscrire sur le site.

III.b.4. Séquence



Ce diagramme montre les étapes du parcours d'une annonce, de sa création à sa suppression une fois que l'objet annoncé a bien été donné.



Ce diagramme montre les étapes de l'inscription de l'utilisateur.

III.c. Base de données

Note :

Les données de la base ayant changé entre la conception et la réalisation, certaines d'entre elles peuvent avoir été déplacées ou supprimées.

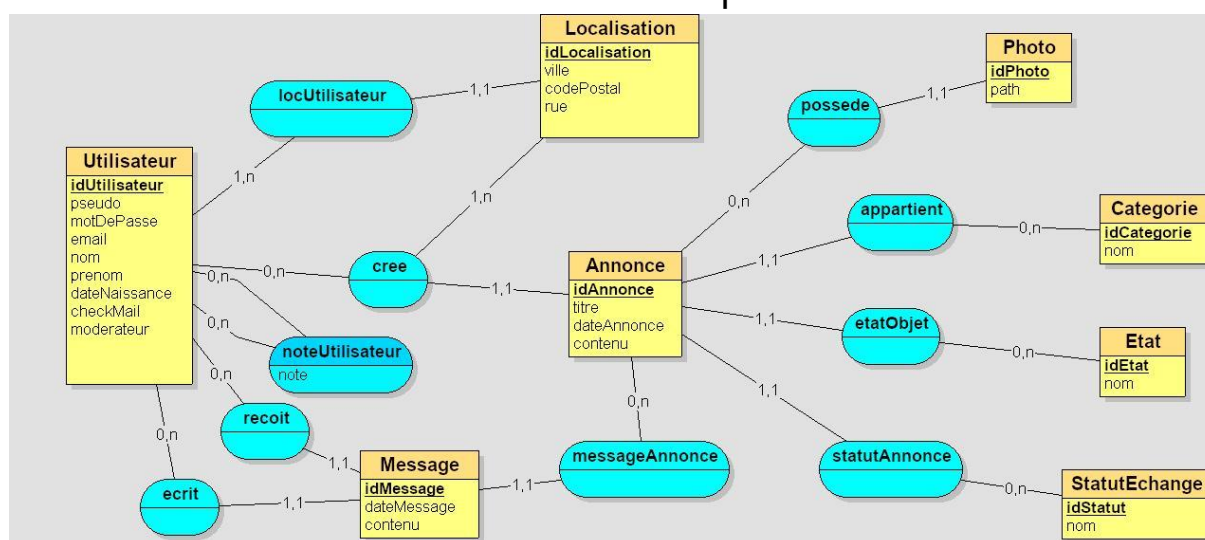
III.b.1. Dictionnaire des données

Le dictionnaire de données ci-dessous liste toutes les entrées qui nécessitent d'être contenues dans la base de données :

NOM	SIGNIFICATION	TYPE	TAILLE	CONTRAINTES
Entité utilisateur				
idUtilisateur	Identifiant de l'utilisateur	int		Clé primaire Auto-incrémenté
pseudo	Pseudonyme de l'utilisateur	varchar	20	Non nul
mdp	Mot de passe de l'utilisateur	varchar	256	Non nul
email	Mail de l'utilisateur	nvarchar	320	Non nul
nom	Nom de l'utilisateur	varchar	50	Non nul
prenom	Prénom de l'utilisateur	varchar	50	Non nul
dateNaissance	Date de naissance de l'utilisateur	date		Non nul
checkMail	Vérification de la validité du mail de l'utilisateur	boolean		Non nul False par défaut
note	Note attribuée à l'utilisateur	int		Non nul Non signé 0 par défaut
role	Droits de l'utilisateur	longtext		Tableau JSON
Entité annonce				
idAnnonce	Identifiant de l'annonce	int		Clé primaire Auto-incrémenté
titreAnnonce	Titre de l'annonce	varchar	128	Non nul
dateAnnonce	Date de publication de l'annonce	datetime		Non nul Now() par défaut
contenuAnnonce	Contenu de l'annonce	longtext		
Entité localisation				
idLocalisation	Identifiant de la localisation	int		Clé primaire Auto-incrémenté
ville	Ville enregistrée	varchar	20	Non nul
codePostal	Code postal de la ville	char	5	Non nul
rue	Voie enregistrée	varchar	256	Non nul
Entité photo				
idPhoto	Identifiant de la photo	int		Clé primaire Auto-incrémenté
pathPhoto	Lien de la photo	text		Non nul
Entité catégorie				
idCategorie	Identifiant de la catégorie	int		Clé primaire Auto-incrémenté

nomCategorie	Nom de la catégorie	varchar	64	Non nul
Entité état				
idEtat	Identifiant de l'état	int		Clé primaire Auto-incrémenté
nomEtat	Nom de l'état	varchar	10	Non nul
Entité statut d'échange				
idStatut	Identifiant du statut d'échange	int		Clé primaire Auto-incrémenté
nomStatut	Nom du statut	varchar	32	Non nul
Entité message				
idMessage	Identifiant du message	int		Clé primaire Auto-incrémenté
dateMessage	Date d'envoi du message	datetime		Non nul Now() par défaut
contenuMessage	Contenu du message	longtext		Non nul

III.b.2. Modèle conceptuel de données

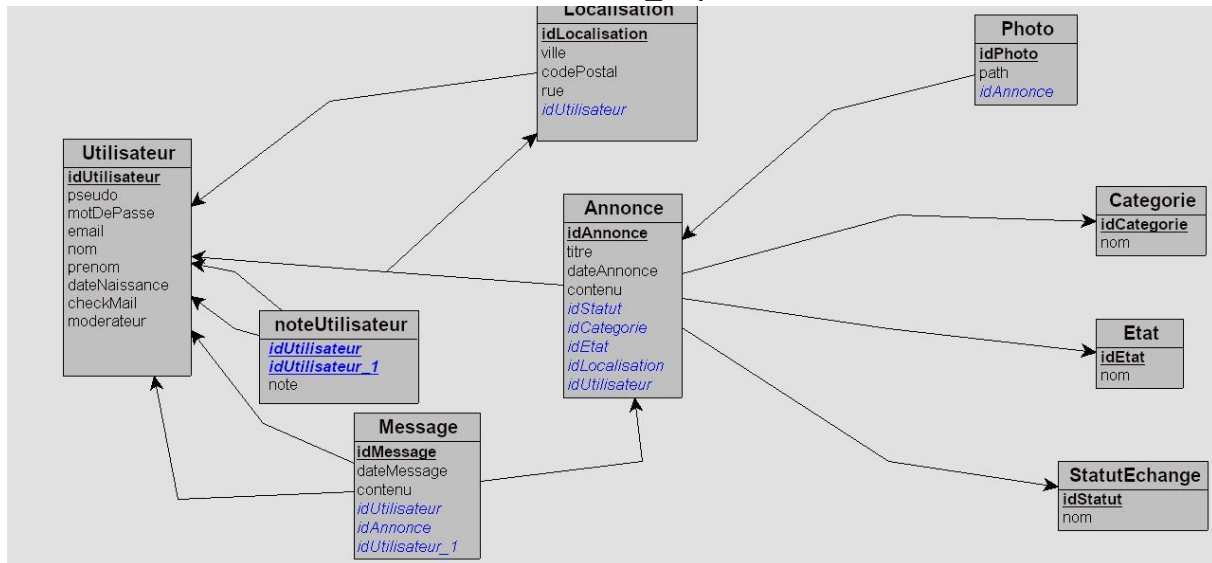


Ce schéma représente les différentes associations entre les entités nécessaires au fonctionnement du site. Les liaisons sont les suivantes :

- Un état est lié à aucune, une ou plusieurs annonce.s (occasion, neuf...),
- Une catégorie est appliquée à aucune, une ou plusieurs annonce.s (que ce soit un jouet, un outil...),
- Un statut est appliqué à aucune, une ou plusieurs annonce.s (en cours de don, à donner, donné),
- Une annonce dispose d'aucune, une ou plusieurs photo.s,
- Un utilisateur a publié aucune, une ou plusieurs annonce.s,

- L'annonce et son auteur partagent la même localisation, mais ce dernier doit en choisir une s'il a plusieurs localisations enregistrées dans son profil (dans ce cas, il s'agit d'une association ternaire annonce-utilisateur-localisation),
- Deux utilisateurs s'envoient des messages entre eux (l'un envoie à l'autre, qui le reçoit), mais ces messages ne se réfèrent toujours qu'à une seule et unique annonce,
- Aucun, un ou plusieurs utilisateur.s peuvent noter aucun, un ou plusieurs autre.s utilisateur.s (cette association est donc réflexive).

III.b.3. Modèle Logique de données



Ce modèle donne une indication de toutes les entrées nécessitant d'être implémentées dans la base de données afin de pouvoir établir son fonctionnement côté serveur. Il contient théoriquement les mêmes informations que le MCD, mais fait apparaître les tables et leurs clés étrangères qui garantissent l'intégrité référentielle entre deux tables.

III.b.4. Script de création de la base

Le script suivant permet de créer la base de données en cohérence avec les modèles des sections précédentes :

```
CREATE DATABASE IF NOT EXISTS Cyclic;
USE Cyclic;
```

```
CREATE TABLE IF NOT EXISTS Categorie(
  idCategorie INT AUTO_INCREMENT,
  nom VARCHAR(50),
  PRIMARY KEY(idCategorie)
);
```

```
CREATE TABLE IF NOT EXISTS Etat(
  idEtat INT AUTO_INCREMENT,
  nom VARCHAR(50),
  PRIMARY KEY(idEtat)
);
```

```

CREATE TABLE IF NOT EXISTS StatutEchange(
    idStatut INT AUTO_INCREMENT,
    nom VARCHAR(50),
    PRIMARY KEY(idStatut)
);

CREATE TABLE IF NOT EXISTS Utilisateur(
    idUtilisateur INT AUTO_INCREMENT,
    pseudo VARCHAR(50) NOT NULL,
    motDePasse VARCHAR(255) NOT NULL,
    email NVARCHAR(320) NOT NULL,
    nom VARCHAR(50) NOT NULL,
    prenom VARCHAR(50) NOT NULL,
    dateNaissance DATE NOT NULL,
    checkMail BOOLEAN NOT NULL DEFAULTS 0,
    modérateur LONGTEXT,
    PRIMARY KEY(idUtilisateur)
);

CREATE TABLE IF NOT EXISTS Localisation(
    idLocalisation INT AUTO_INCREMENT,
    ville VARCHAR(256) NOT NULL,
    codePostal VARCHAR(5) NOT NULL,
    rue VARCHAR(20) NOT NULL,
    idUtilisateur INT NOT NULL,
    PRIMARY KEY(idLocalisation),
    FOREIGN KEY(idUtilisateur) REFERENCES Utilisateur(idUtilisateur)
);

CREATE TABLE IF NOT EXISTS Annonce(
    idAnnonce INT AUTO_INCREMENT,
    titre VARCHAR(50) NOT NULL,
    dateAnnonce DATETIME NOT NULL,
    contenu TEXT,
    idStatut INT NOT NULL,
    idCategorie INT NOT NULL,
    idEtat INT NOT NULL,
    idLocalisation INT NOT NULL,
    idUtilisateur INT NOT NULL,
    PRIMARY KEY(idAnnonce),
    FOREIGN KEY(idStatut) REFERENCES StatutEchange(idStatut),
    FOREIGN KEY(idCategorie) REFERENCES Categorie(idCategorie),
    FOREIGN KEY(idEtat) REFERENCES Etat(idEtat),
    FOREIGN KEY(idLocalisation) REFERENCES Localisation(idLocalisation),
    FOREIGN KEY(idUtilisateur) REFERENCES Utilisateur(idUtilisateur)
);

CREATE TABLE IF NOT EXISTS Photo(
    idPhoto INT AUTO_INCREMENT,
    path VARCHAR(255) NOT NULL,

```

```

        idAnnonce INT NOT NULL,
        PRIMARY KEY(idPhoto),
        FOREIGN KEY(idAnnonce) REFERENCES Annonce(idAnnonce)
    );

CREATE TABLE IF NOT EXISTS Message(
    idMessage INT AUTO_INCREMENT,
    dateMessage DATETIME NOT NULL,
    contenu TEXT NOT NULL,
    idUtilisateur INT NOT NULL,
    idAnnonce INT NOT NULL,
    idUtilisateur_1 INT NOT NULL,
    PRIMARY KEY(idMessage),
    FOREIGN KEY(idUtilisateur) REFERENCES Utilisateur(idUtilisateur),
    FOREIGN KEY(idAnnonce) REFERENCES Annonce(idAnnonce),
    FOREIGN KEY(idUtilisateur_1) REFERENCES Utilisateur(idUtilisateur)
);

CREATE TABLE IF NOT EXISTS noteUtilisateur(
    idUtilisateur INT,
    idUtilisateur_1 INT,
    note INT UNSIGNED NOT NULL DEFAULTS 0,
    PRIMARY KEY(idUtilisateur, idUtilisateur_1),
    FOREIGN KEY(idUtilisateur) REFERENCES Utilisateur(idUtilisateur),
    FOREIGN KEY(idUtilisateur_1) REFERENCES Utilisateur(idUtilisateur)
);

```

Note :

La base de données sera créée via Symfony, ce script n'est donc qu'une représentation de la façon dont elle aurait été réalisée via MySQL.