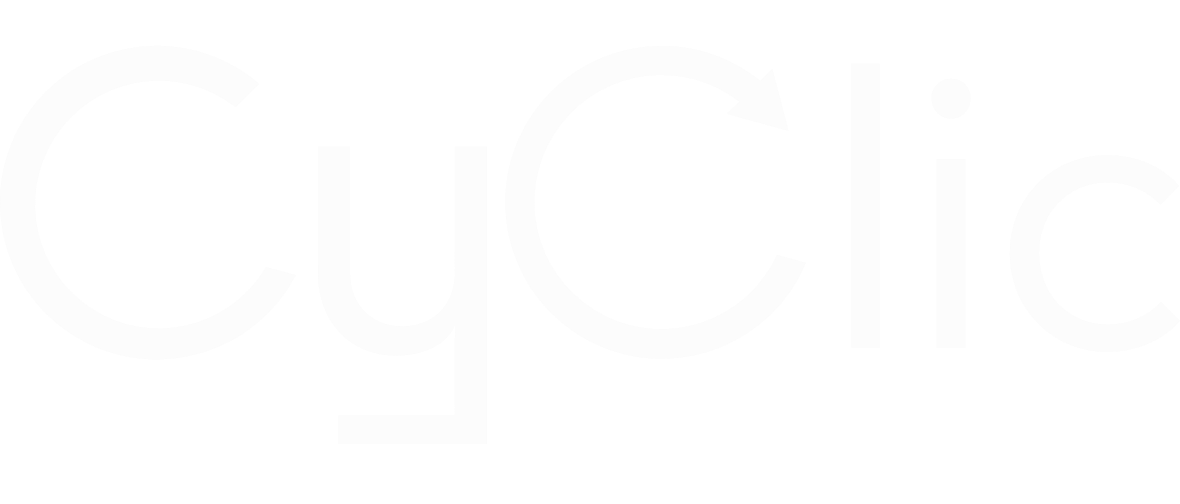
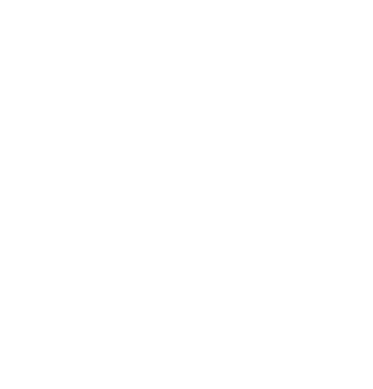
**Dossier d’analyse**





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

**Table des matières**

[I. Présentation 2](#_Toc92955274)

[I.a. Qu’est-ce que CyClic ? 2](#_Toc92955275)

[I.b. Objectifs du document d’analyse 2](#_Toc92955276)

[II. Acteurs 2](#_Toc92955277)

[III. Diagrammes 3](#_Toc92955278)

[III.a. Classes 3](#_Toc92955279)

[III.b. Cas d’utilisation 4](#_Toc92955280)

[III.c. Etat-transition 5](#_Toc92955281)

[III.d. Séquence 8](#_Toc92955282)

[IV. Base de données 10](#_Toc92955283)

[IV.a. Dictionnaire des données 10](#_Toc92955284)

[IV.b. Modèle conceptuel de données 12](#_Toc92955285)

[IV.c. Modèle logique de données 13](#_Toc92955286)

[IV.d. Script SQL de création de la BdD 13](#_Toc92955287)

**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 |
| 2022.01.10 | l’équipe | 0.01b | Reconversion en dossier d’analyse |
| 2O22.01.12 | Ledda Damien | 0.02b | Ajout d’images |

1. Présentation

## 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 d’analyse

Ce document a pour but de présenter les différents diagrammes et modèles pouvant justifier le fonctionnement du site.

1. Acteurs

Chez CyClic, les acteurs sont répartis en 3 catégories :

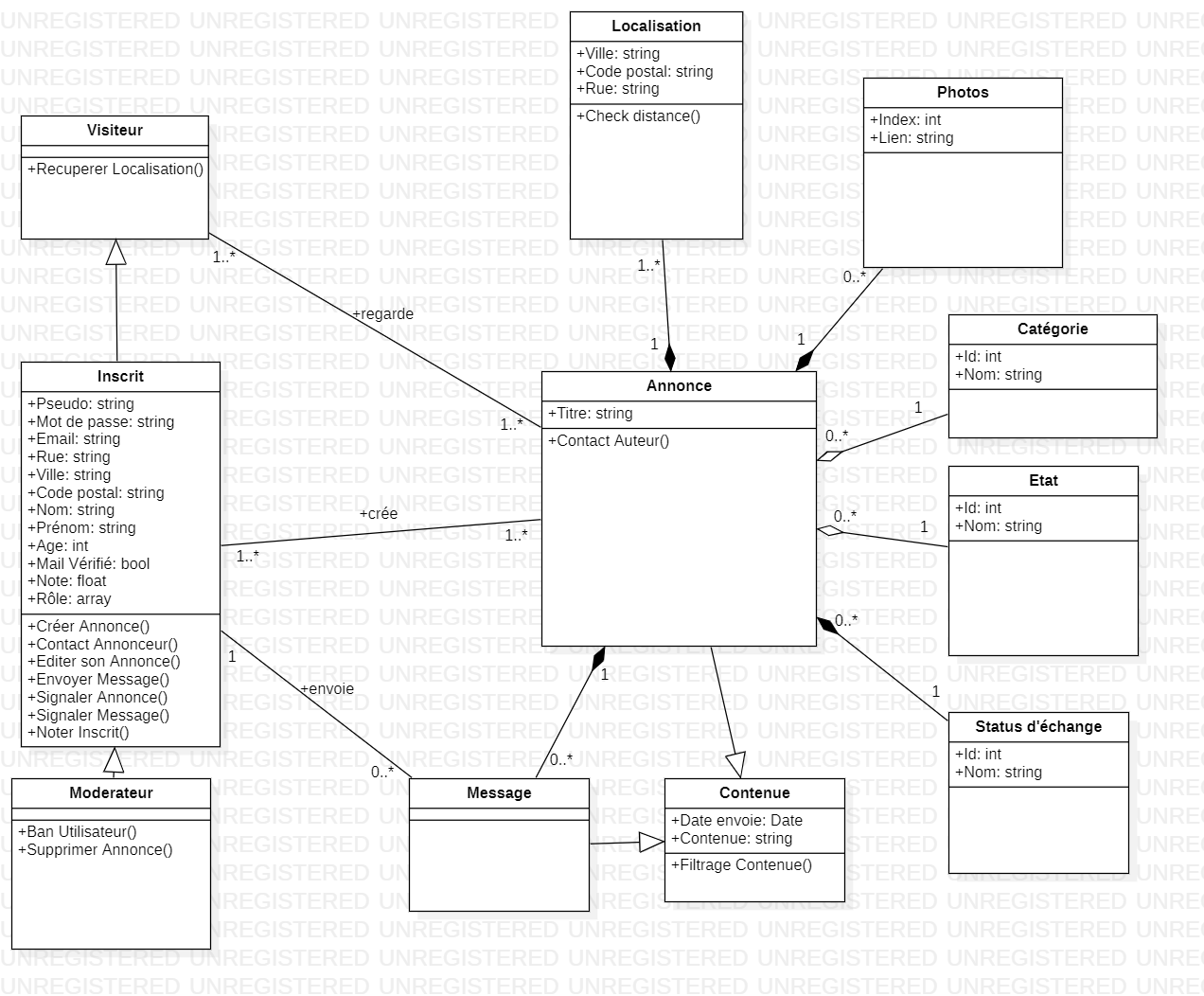
* Le **visiteur**, qui peut :
  + Visiter le site et faire une recherche d’annonces
  + Consulter une annonce précise qui lui aurait tapé dans l’œil.

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 :
  + Créer une annonce concernant un objet dont il n’a plus besoin
  + Interagir avec des annonces d’autres utilisateurs
  + Envoyer des messages et prendre rendez-vous avec un autre utilisateur afin de réaliser un don
  + Noter un utilisateur après un échange.
* Le **modérateur**/l’**administrateur**, héritant des droits d’un utilisateur inscrit, a en plus la possibilité de :
  + Gérer les catégories (en ajouter, en supprimer, en éditer) et les états
  + Supprimer les annonces qui ne sont pas conformes aux règles du site
  + Contrôler les signalements d’utilisateurs (et, par extension, bannir les utilisateurs abusifs)

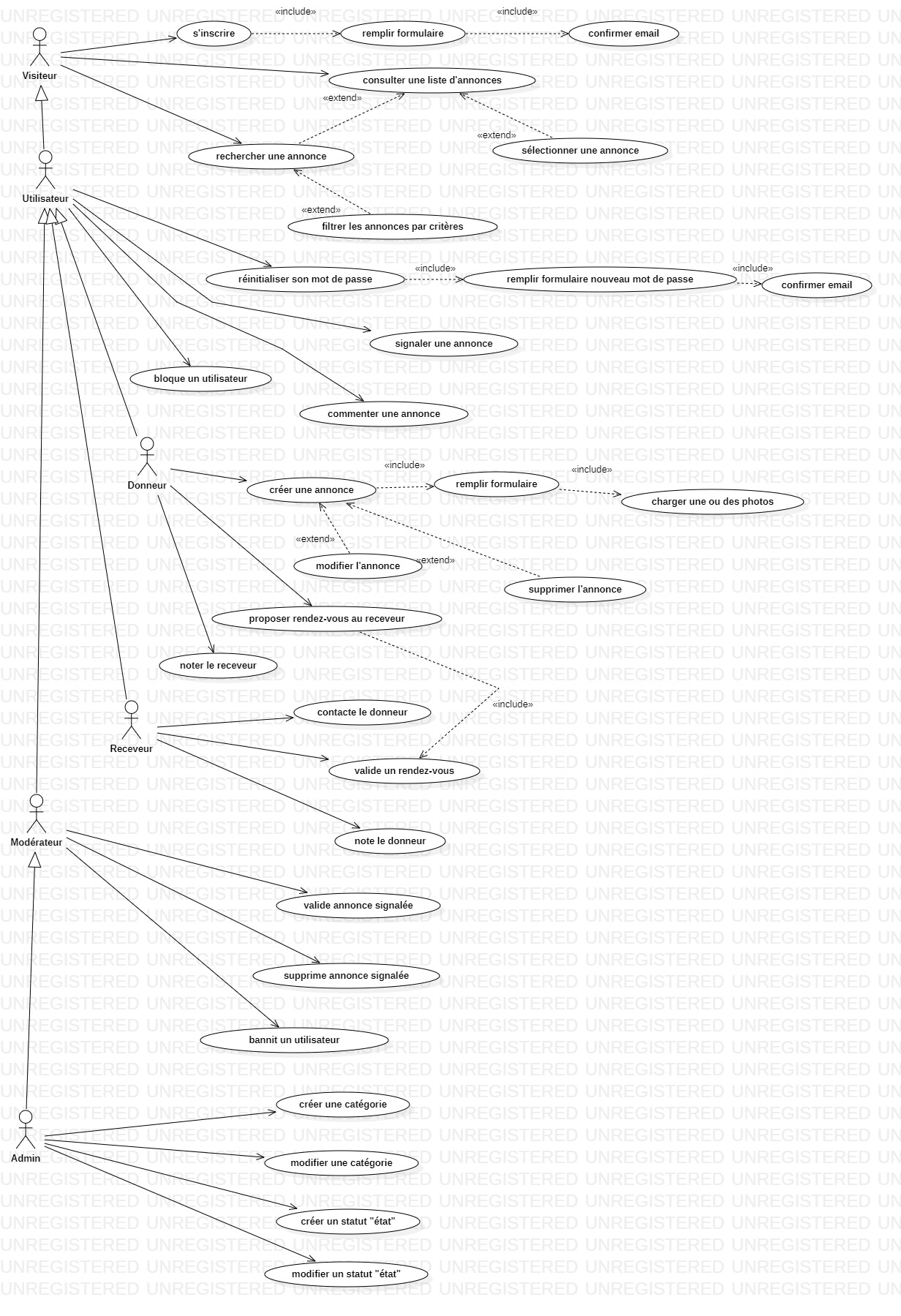
1. Diagrammes

## III.a. Classes



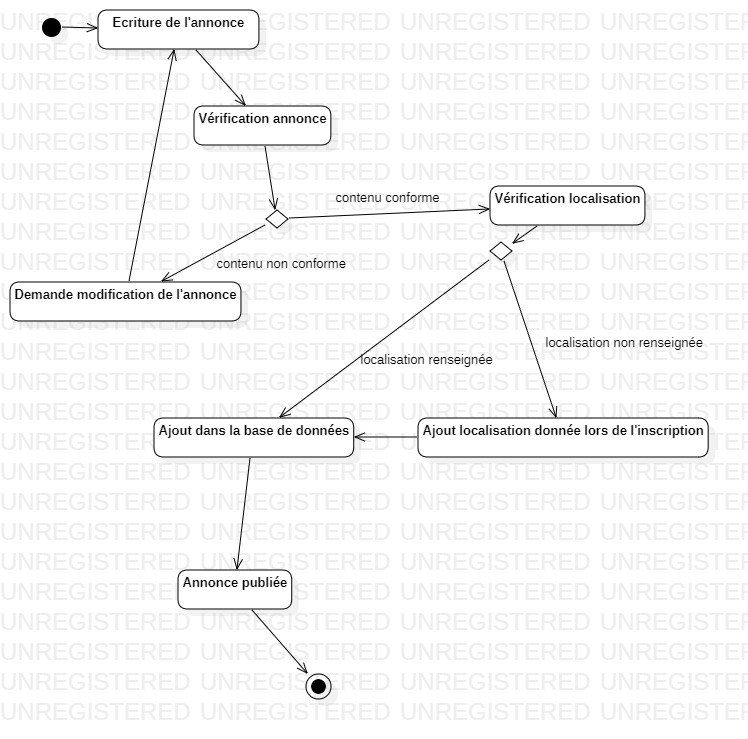
Le diagramme de classe montre les différentes possibilités d’action offertes à l’utilisateur en fonction de son statut. Il présente également les différents attributs liés aux différentes entités ainsi que les liens d’héritage (un modérateur hérite des caractéristiques d’un inscrit) et de possession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

## III.b. Cas d’utilisation

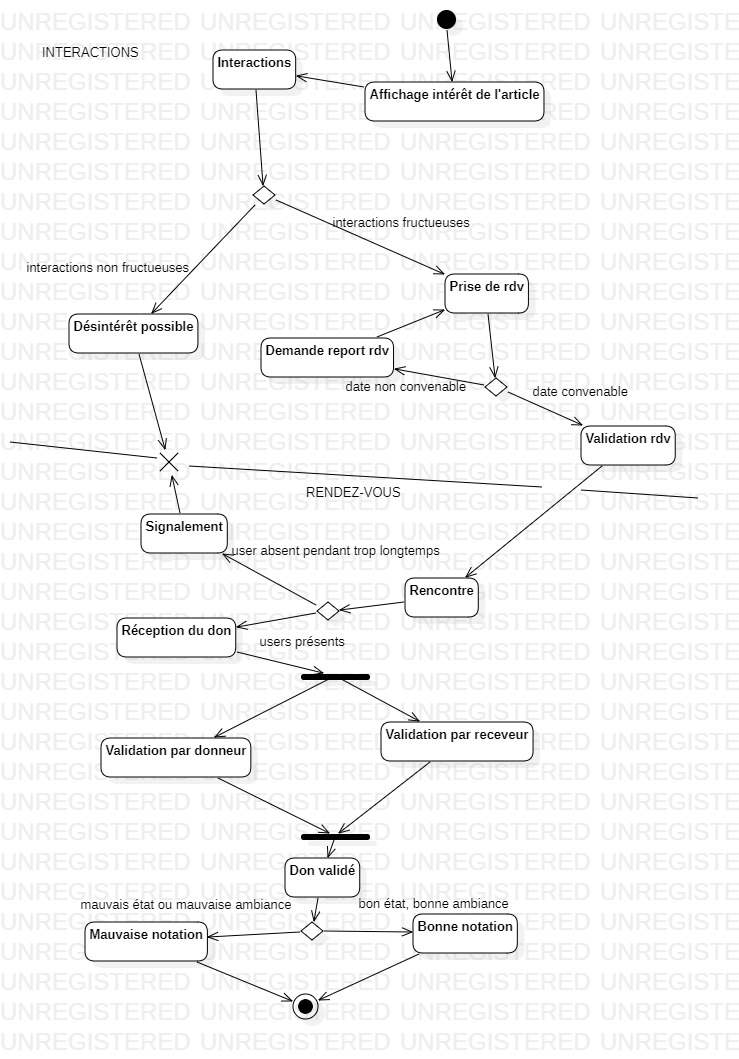


Ce diagramme de cas d’utilisation présente toutes les actions que peut réaliser un utilisateur en fonction de son statut/rôle.

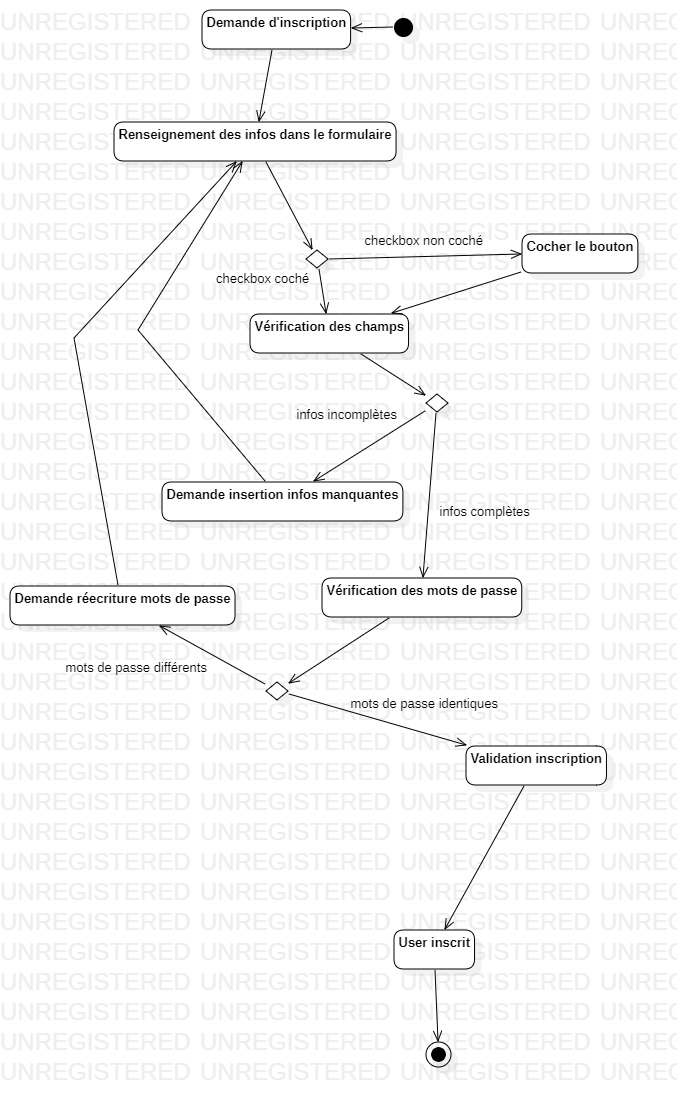
## III.c. Etat-transition



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

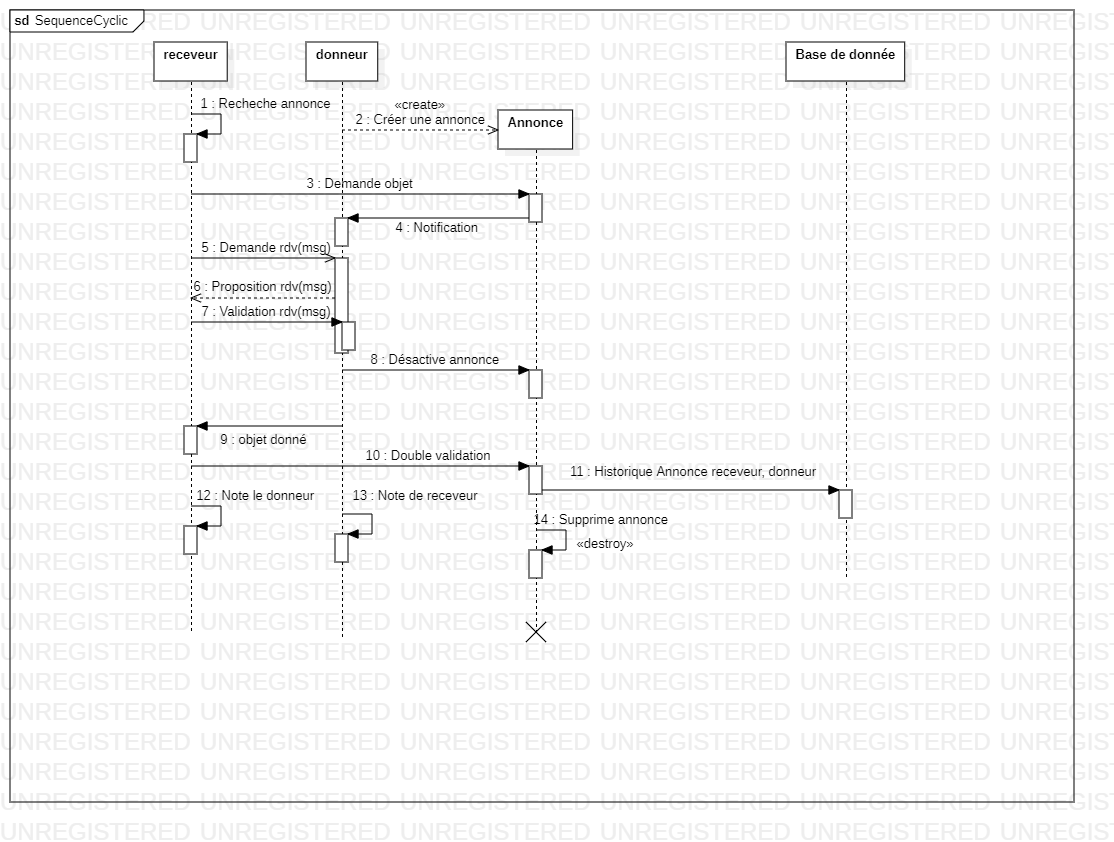


Ce diagramme montre les différents états et transitions de la prise de contact d’un donneur, à la remise de l’objet qui finit par la notation entre utilisateur.



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

## III.d. 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.

Une image contenant table

Description générée automatiquement

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

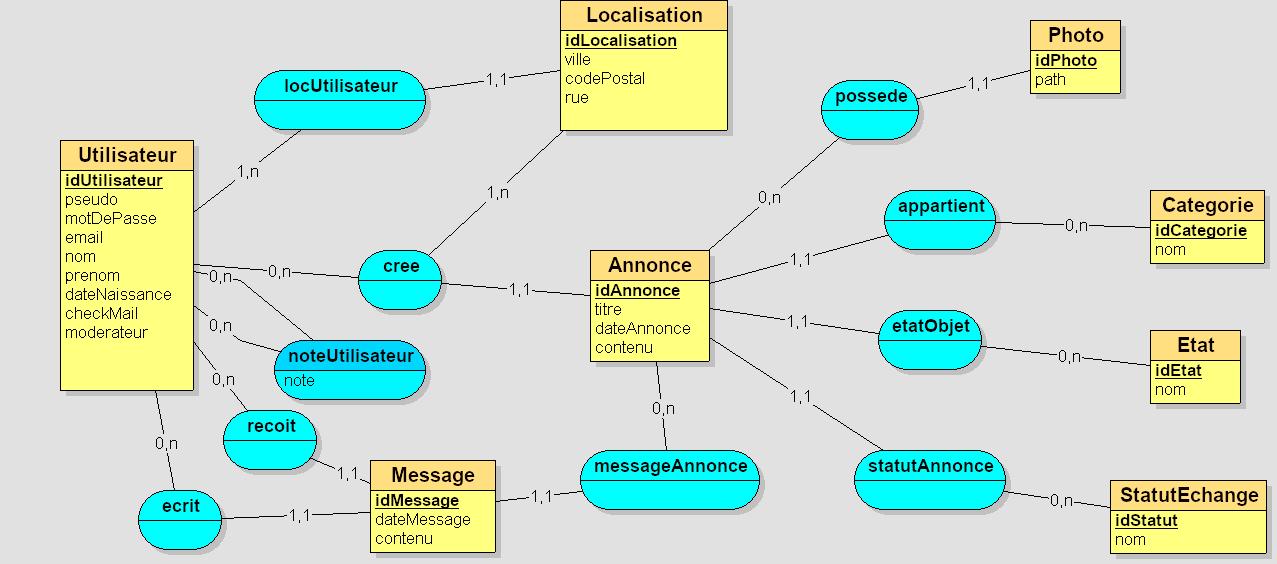
1. Base de données

## IV.a. 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’utiliasteur | boolean |  | Non nul  False par défaut |
| note | Note attribuée à l’utilisateur | int |  | Non nul  Non signé  0 par défaut |
| moderateur | Droits additionnels de l’utilisateur (en JSON) | longtext |  |  |
| 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 |

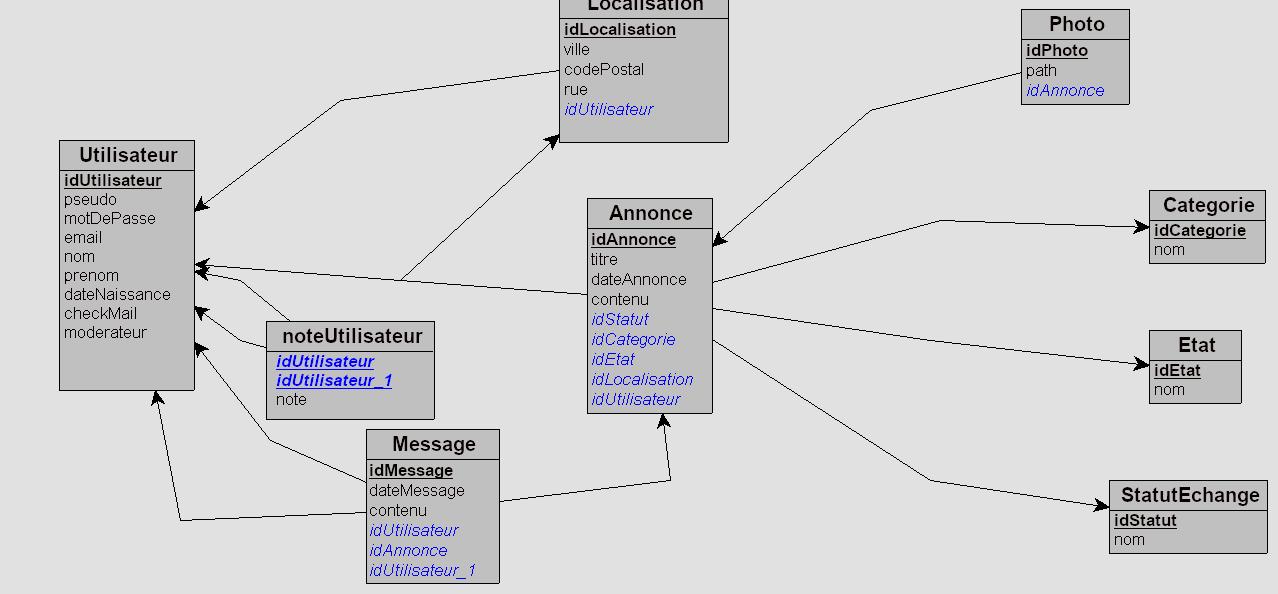
## IV.b. Modèle conceptuel de données



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 peu.ven.t noter aucun, un ou plusieurs autre.s utilisateur.s (cette association est donc réflexive).

## IV.c. 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.

## IV.d. Script SQL de création de la BdD

Le script suivant permet de créer la base de données en cohérence avec les modèles des sections IV.b et IV.c et du dictionnaire de données de la section IV.a :

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,

moderateur 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)

);