

LA2 - POO - GeoCache

Baptiste Pautrat & Gaston Deseine

Introduction

Sujet : Développer une application java console à destinations de testeurs permettant de valider la robustesse et la souplesse de la couche d'accès aux données d'une application de gestion de GeoCaches.

Cf. : cahier des charges.

Technologies utilisées

- Java 15.0.2
- Hibernate 5.4
- JPA 2.0
- MySQL 5.7

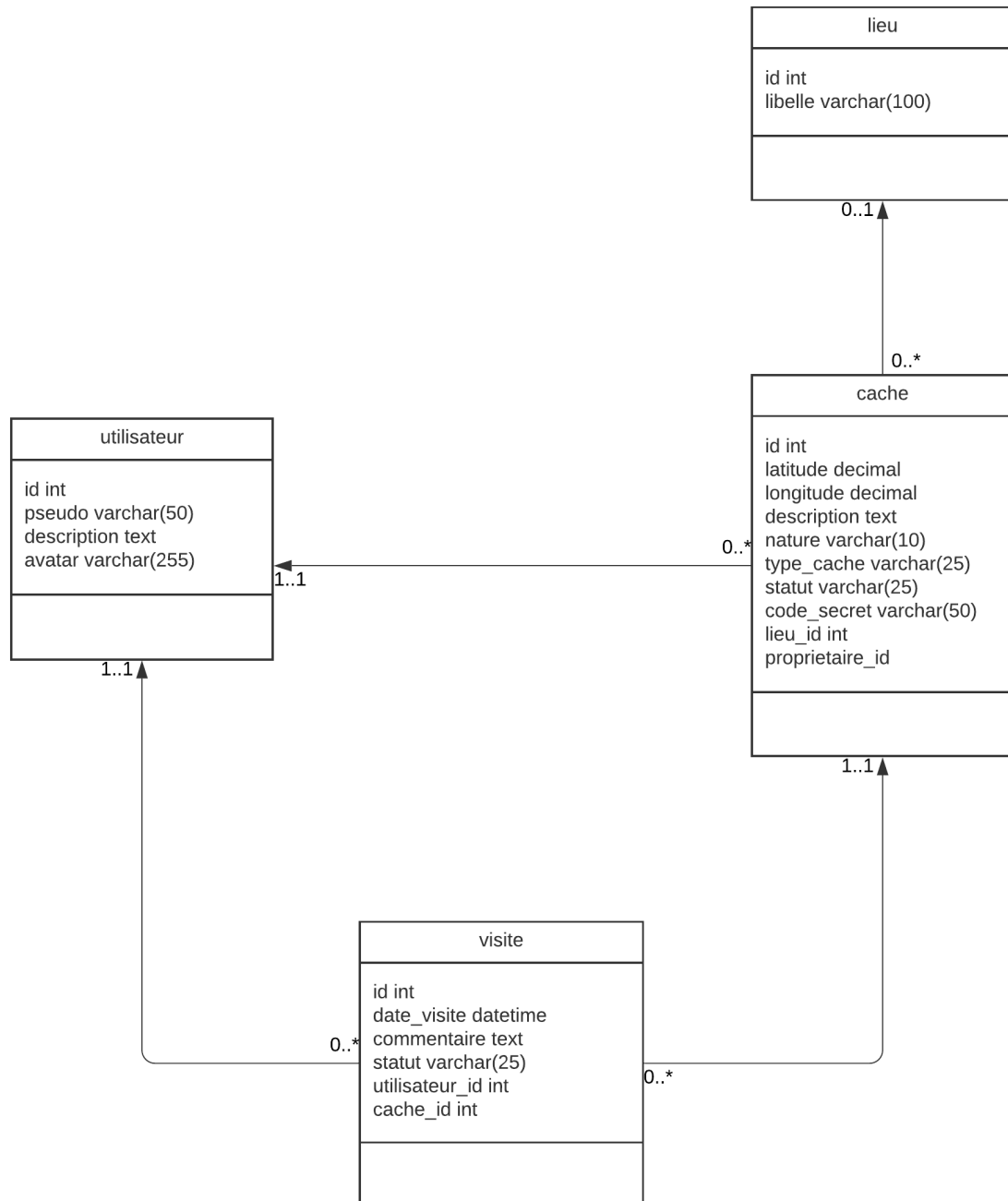
Conception

Architecture de base de données

Tables :

- Lieu : un lieu géographique définit par un libellé.
- Cache : informations permettant de situer et définir une cache.
- Utilisateur : information d'un utilisateur.
- Visite : lien entre un utilisateur et une cache.

Modèle UML :



SQL DDL

```

create database la2-geocache;
use la2-geocache;

create table cache
(
    id int auto_increment
        primary key,
    latitude decimal null,
    longitude decimal null,
    description text null,
    nature varchar(10) not null,
    type_cache varchar(25) default 'traditionnelle' not null,
    statut varchar(25) default 'inactive' not null,
    code_secret varchar(50) not null,
    lieu_id int null,
    proprietaire_id int not null

```

```

);

create index cache_lieu_id_fk
  on cache (lieu_id);

create index cache_utilisateur_id_fk
  on cache (proprietaire_id);

create table lieu
(
  libelle varchar(100) not null,
  id int auto_increment
      primary key
);

create table utilisateur
(
  id int auto_increment
      primary key,
  pseudo varchar(50) not null,
  description text null,
  avatar varchar(255) default 'default.png' not null,
  constraint utilisateur_pseudo_uindex
      unique (pseudo)
);

create table visite
(
  id int auto_increment
      primary key,
  date_visite datetime null,
  utilisateur_id int not null,
  cache_id int not null,
  commentaire text null,
  statut varchar(25) default 'En cours' not null
);

create index visite_cache_id_fk
  on visite (cache_id);

create index visite_utilisateur_id_fk
  on visite (utilisateur_id);

```

Nos choix

Voici les choix que nous avons réalisés pour le moment :

- Code secret associé a une cache qui permet de valider la visite par un utilisateur. Si le code n'est pas bon la visite n'est pas comptabilisé. Cela constitue une preuve de la visite.

- On stock l'avatar d'un utilisateur dans un répertoire. On indique le chemin vers l'avatar de l'utilisateur dans la base de donnée en temps que VARCHAR(255). Le fichier de l'avatar est nommé en concaténant le pseudo de l'utilisateur (unique) avec l'extension du fichier. Une image par défaut est attribué à l'utilisateur tant qu'il n'a pas défini la sienne.
- Nous avons fait le choix de réduire l'état d'une cache a inactif ou actif. Nous avons trouvé que en cours d'activation, fermée et suspendue était des sous catégories d'inactif. Nous avons jugé que nous n'avions pas besoin de ce niveau d'information.