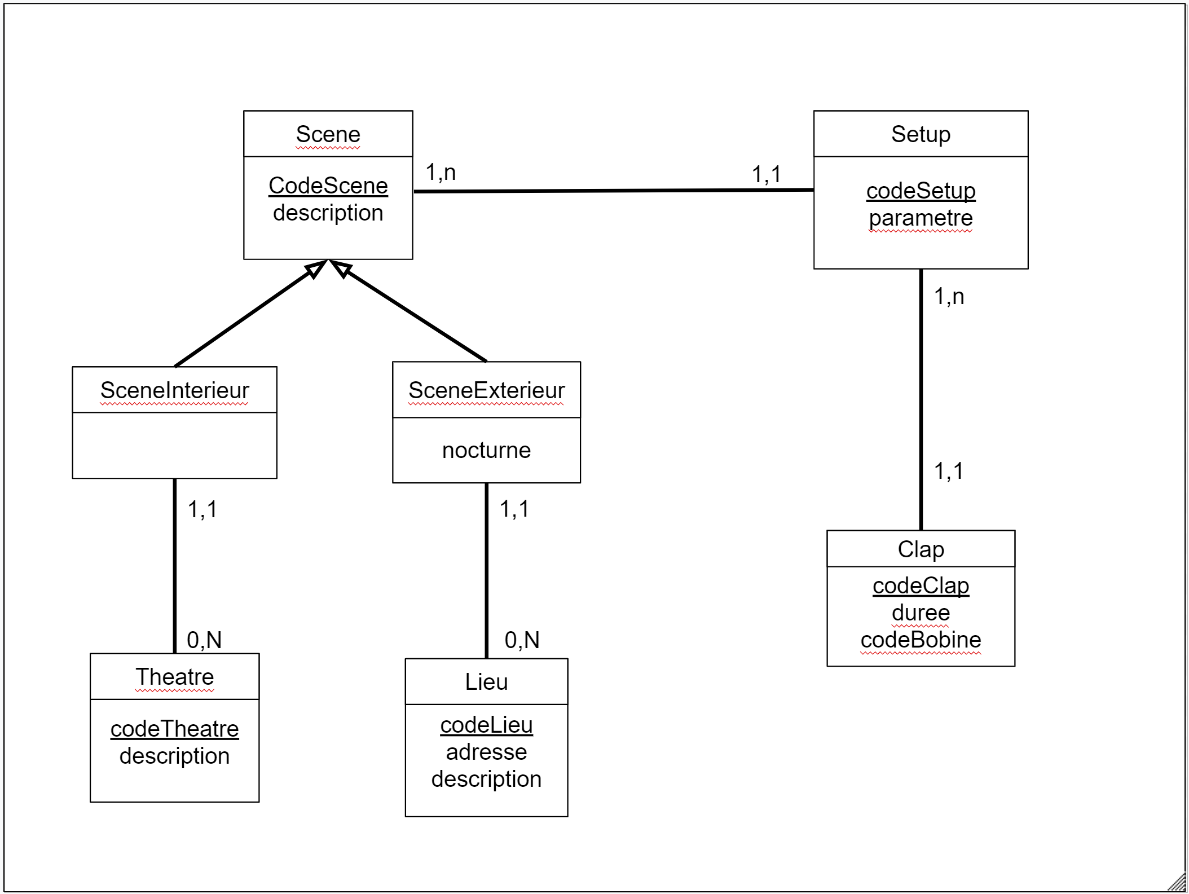
Rendu

Pour ce projet nous avons utilisé l’architecture MVC (Modèle-vue-contrôleur).

Diagramme de classe (complet)

Nous avons utilisé ce Modèle Conceptuel de données :



Nous avons donc pour ce projet utilisé la library hibernate.

Pour gérer la durée d’un Clap, nous avons utilisé en java le type Duration et pour gérer la variable nocture de la SceneExterieur, nous avons utilisé un integer, 0 qui correspond au jour et 1 qui correspond à la nuit.

Notre modèle relationnel de données :

Scene (codeScene, descritpion)

SceneInterieur (#codeTheatre)

SceneExterieur (nocturne, #codeLieu)

Setup (codeSetup, parametre, #codeScene)

Clap (codeClap, duree, codeBobine, #codeSetup)

Lieu (codeLieu, adresse, description)

Theatre (codeTheatre, description)

Les différentes fonctionnalités que nous avons implémentées :

* Créer, modifier, supprimer un théâtre
* Créer, modifier, supprimer un lieu
* Créer, modifier, supprimer un clap
* Créer, modifier, supprimer un setup
* Créer, modifier, supprimer une scène extérieure
* Créer, modifier, supprimer une scène intérieure
* Etant donné une scène, retourner le temps total écoulé pour tous les claps de tous les setups du film
* Etant donné un setup, retourner le temps total écoulé pour tous les claps de ce setup
* Etant donné un setup, retourner les numéros de toutes les bobines du setup
* Etant donnée une scène, retourner le code du lieu de la scène pour celles en plein air et le code du théâtre pour celle à l’intérieur

Nous avons fait en sorte que lorsqu’un utilisateur ajoute un clap à un setup alors le clap est ajouté à la base de données. De même pour l’ajout d’un setup à une scène.

Pour la suppression, nous avons paramétrer hibernate en cascade. Donc si l’utilisateur supprime une scène, alors tous les setups de la scène sont supprimés et tous les claps de tous les setups sont supprimés.