

Spécification

Introduction

Use Case Diagram

Component Diagram

Sequence Diagram (Authentification)

Sequence Diagram (Publication d'un avis)

Package Diagram

Modèle Conceptuel de Données

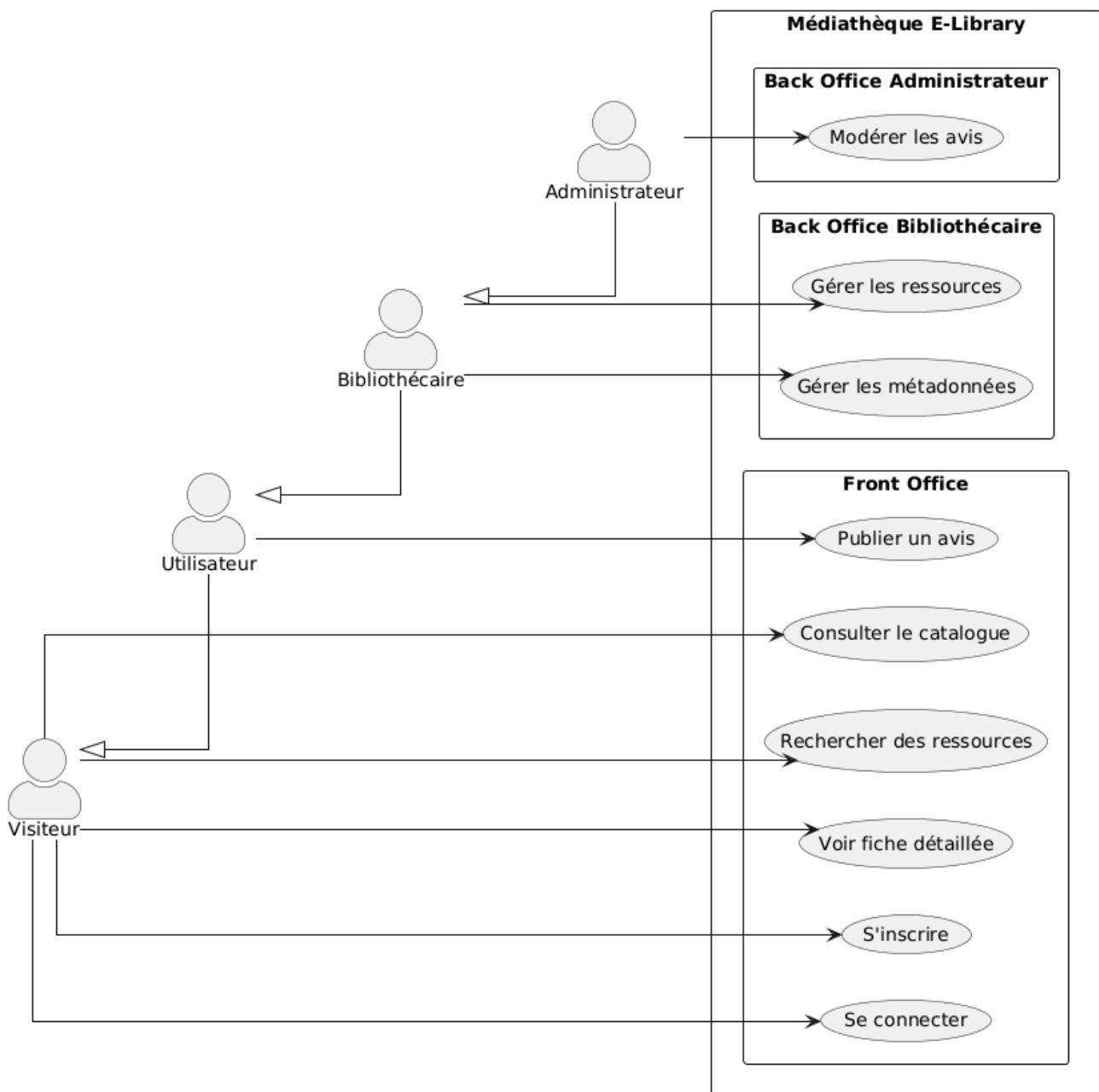
Modèle Logique de Données

Script SQL de création de la base de données

Use Case Diagram (diagramme de cas d'utilisation)

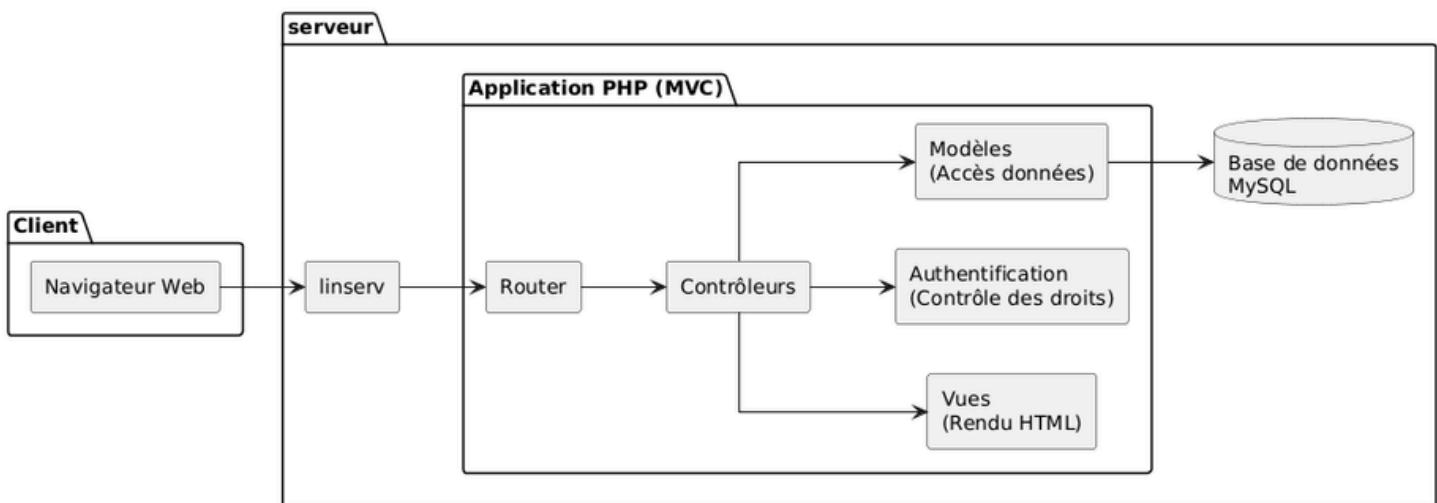
Le diagramme de cas d'utilisation présente les fonctionnalités principales offertes par la médiathèque numérique.

Il met en évidence les différents acteurs du système (utilisateur, bibliothécaire, administrateur) ainsi que les actions qu'ils peuvent effectuer selon leurs droits. Ce diagramme permet d'avoir une vue fonctionnelle globale du système, sans entrer dans les détails techniques.



Component Diagram

Le diagramme de composants illustre l'architecture globale de l'application web. Il montre les différents composants logiciels (navigateur, application PHP, base de données MySQL, ressources statiques) et les interactions entre eux. Ce diagramme permet de comprendre comment les différentes briques techniques coopèrent pour assurer le fonctionnement du système.

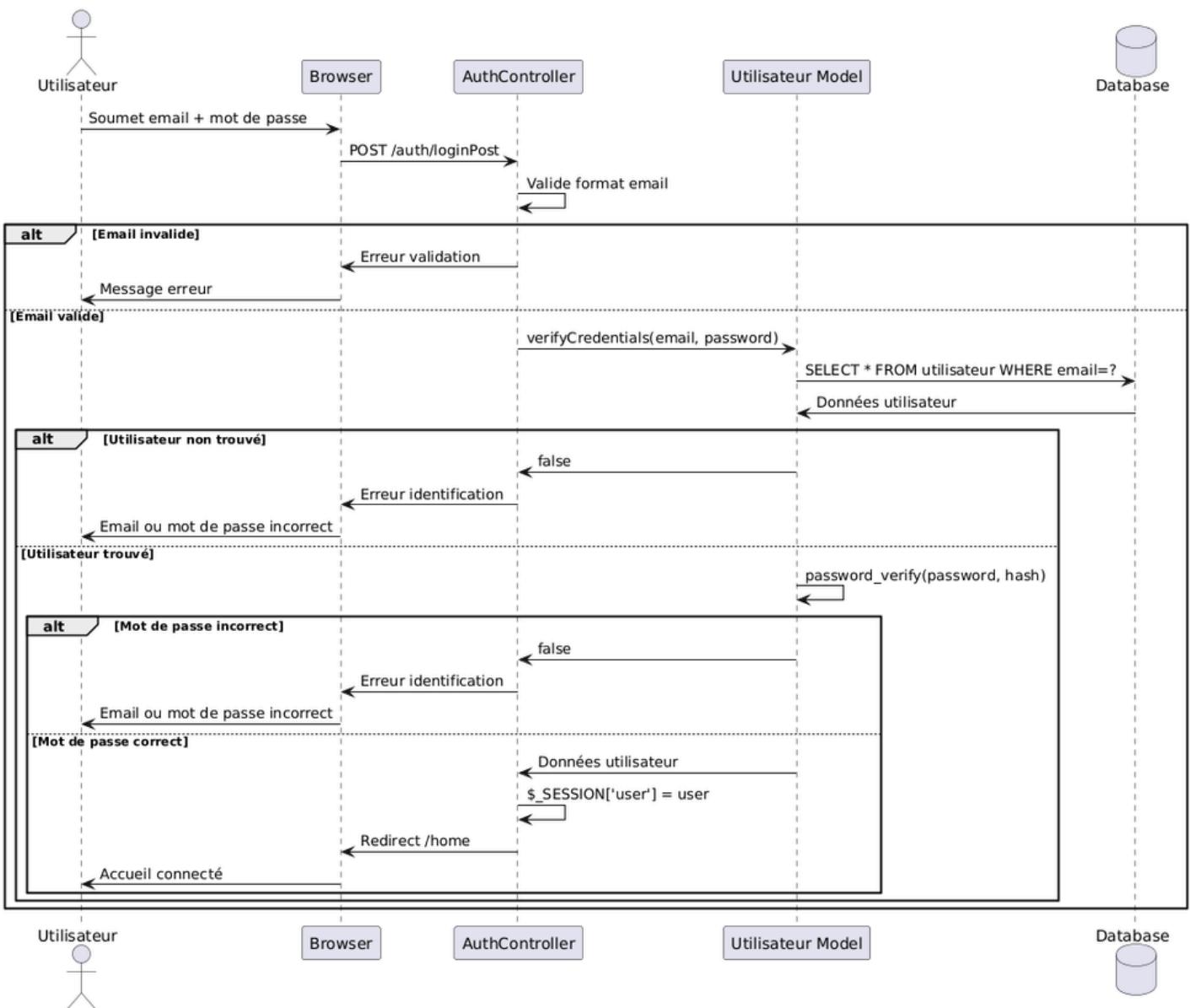


Sequence Diagram (Authentification)

Ce diagramme de séquence illustre le processus d'authentification d'un utilisateur.

Il montre la validation des identifiants, l'accès à la base de données et la création de la session utilisateur.

Ce diagramme permet de comprendre comment la sécurité et l'accès au système sont gérés.

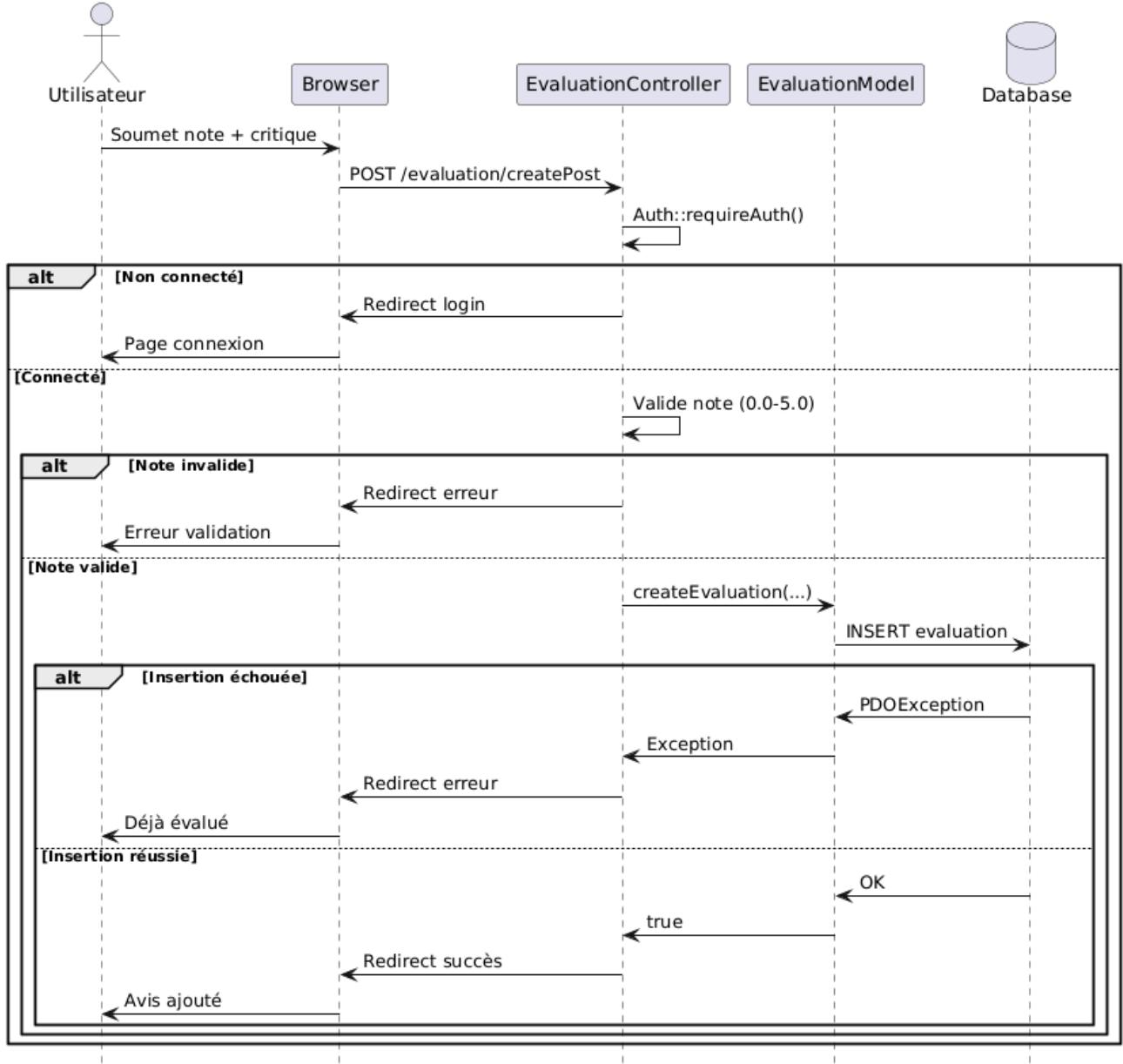


Sequence Diagram : (Publication d'un avis)

Ce diagramme de séquence décrit la publication d'un avis par un utilisateur connecté sur une ressource.

Il met en évidence les contrôles effectués (authentification, validation des données, prévention des doublons) avant l'enregistrement de l'avis en base de données.

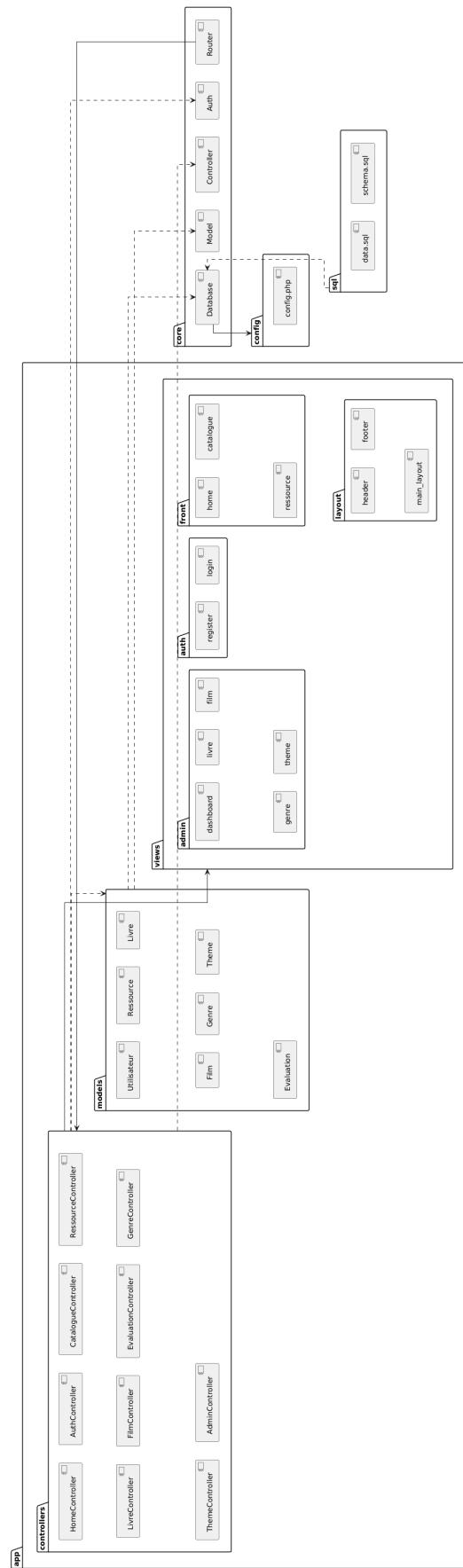
Ce scénario illustre l'interaction complète entre l'interface utilisateur et la logique métier.



Package Diagram

Le diagramme de packages présente l'organisation interne du projet selon l'architecture MVC.

Ce diagramme permet de visualiser clairement la structure du projet et les dépendances entre les différentes couches.

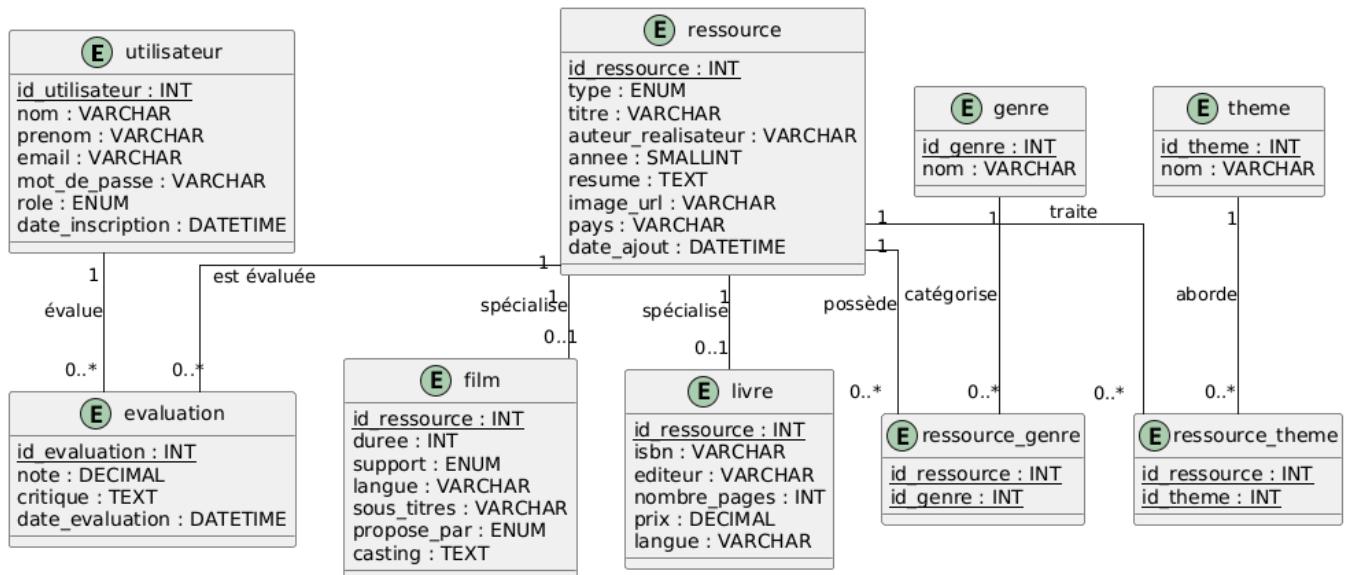


Modèle Conceptuel de Données (MCD)

Le modèle conceptuel de données décrit les entités principales de la médiathèque et les relations qui les lient.

Il représente la vision métier des données, indépendamment de toute considération technique ou d'implémentation.

Ce modèle sert de base à la conception de la base de données relationnelle.

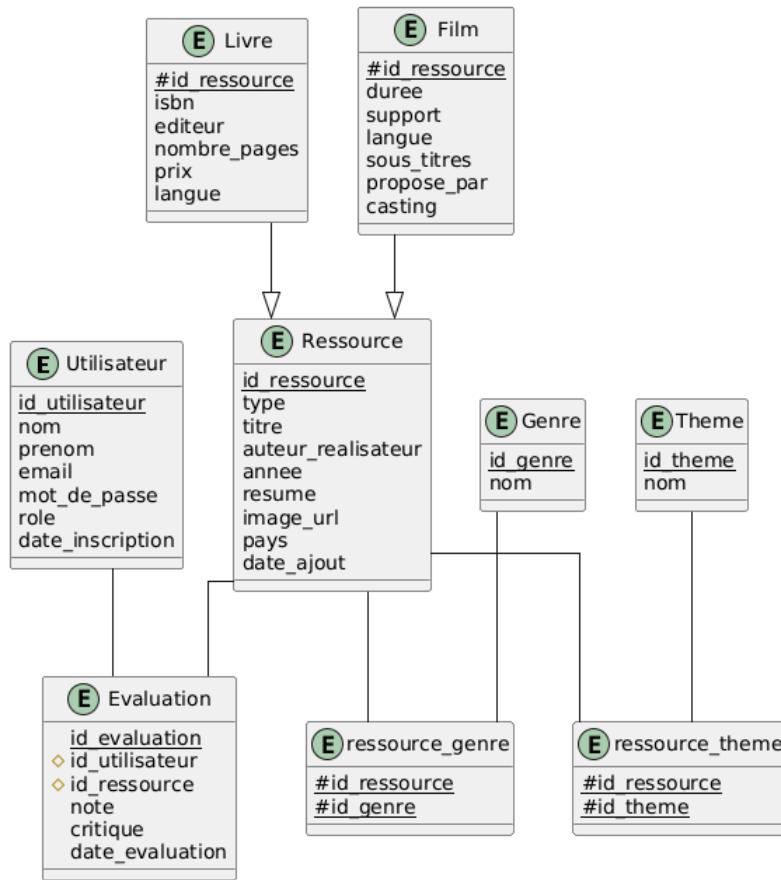


Modèle Logique de Données (MLD)

Le modèle logique de données traduit le MCD en tables relationnelles exploitable par un SGBD.

Il précise les clés primaires, les clés étrangères sans les cardinalités.

Ce modèle constitue une étape intermédiaire avant l'implémentation SQL.



Script SQL de création de la base de données

Le script SQL présenté permet de créer l'ensemble des tables constituant la base de données de la médiathèque.

Il définit la structure relationnelle, les clés primaires, les clés étrangères et les contraintes d'intégrité.

```

DROP DATABASE IF EXISTS mediatheque_sae;
CREATE DATABASE mediatheque_sae CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
USE mediatheque_sae;

DROP TABLE IF EXISTS evaluation;
DROP TABLE IF EXISTS ressource_genre;
DROP TABLE IF EXISTS ressource_theme;
DROP TABLE IF EXISTS film;
DROP TABLE IF EXISTS livre;
DROP TABLE IF EXISTS ressource;
DROP TABLE IF EXISTS genre;
DROP TABLE IF EXISTS theme;
DROP TABLE IF EXISTS utilisateur;

CREATE TABLE utilisateur (
    id_utilisateur INT AUTO_INCREMENT PRIMARY KEY,
    nom VARCHAR(100) NOT NULL,
    prenom VARCHAR(100) NOT NULL,
    email VARCHAR(255) NOT NULL UNIQUE,
    mot_de_passe VARCHAR(255) NOT NULL,
    role ENUM('utilisateur', 'bibliothecaire', 'administrateur') NOT NULL DEFAULT 'utilisateur',
    date_inscription DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    INDEX idx_email (email),
    INDEX idx_role (role)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

CREATE TABLE ressource (
    id_ressource INT AUTO_INCREMENT PRIMARY KEY,
    type ENUM('livre', 'film') NOT NULL,
    titre VARCHAR(255) NOT NULL,
    auteur_realisateur VARCHAR(255) NOT NULL,
    annee SMALLINT NOT NULL,
    resume TEXT,
    image_url VARCHAR(255),
    pays VARCHAR(100),
    date_ajout DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    INDEX idx_type (type),
    INDEX idx_titre (titre),
    INDEX idx_auteur_realisateur (auteur_realisateur),
    INDEX idx_date_ajout (date_ajout)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

CREATE TABLE livre (
    id_ressource INT PRIMARY KEY,
    isbn VARCHAR(13) NOT NULL UNIQUE,
    editeur VARCHAR(255) NOT NULL,
    nombre_pages INT NOT NULL,
    prix DECIMAL(10,2),
    langue VARCHAR(50) NOT NULL DEFAULT 'Français',
    CONSTRAINT fk_livre_ressource FOREIGN KEY (id_ressource)
        REFERENCES ressource(id_ressource)
        ON DELETE CASCADE,
    INDEX idx_isbn (isbn)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

CREATE TABLE film (
    id_ressource INT PRIMARY KEY,
    duree INT NOT NULL,
    support ENUM('DVD', 'Blu-ray', 'Streaming') NOT NULL,
    langue VARCHAR(50) NOT NULL,
    sous_titres VARCHAR(255),
    propose_par ENUM('Arte', 'UniversCiné', 'CNC', 'Médiathèque numérique') NULL,
    casting TEXT,
    CONSTRAINT fk_film_ressource FOREIGN KEY (id_ressource)
        REFERENCES ressource(id_ressource)
        ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

CREATE TABLE theme (
    id_theme INT AUTO_INCREMENT PRIMARY KEY,
    nom VARCHAR(100) NOT NULL UNIQUE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

CREATE TABLE genre (
    id_genre INT AUTO_INCREMENT PRIMARY KEY,
    nom VARCHAR(100) NOT NULL UNIQUE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

CREATE TABLE ressource_theme (
    id_ressource INT NOT NULL,
    id_theme INT NOT NULL,
    PRIMARY KEY (id_ressource, id_theme),
    CONSTRAINT fk_rt_ressource FOREIGN KEY (id_ressource)
        REFERENCES ressource(id_ressource)
        ON DELETE CASCADE,
    CONSTRAINT fk_rt_theme FOREIGN KEY (id_theme)
        REFERENCES theme(id_theme)
        ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

CREATE TABLE ressource_genre (
    id_ressource INT NOT NULL,
    id_genre INT NOT NULL,
    PRIMARY KEY (id_ressource, id_genre),
    CONSTRAINT fk_rg_ressource FOREIGN KEY (id_ressource)
        REFERENCES ressource(id_ressource)
        ON DELETE CASCADE,
    CONSTRAINT fk_rg_genre FOREIGN KEY (id_genre)
        REFERENCES genre(id_genre)
        ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

CREATE TABLE evaluation (
    id_evaluation INT AUTO_INCREMENT PRIMARY KEY,
    id_utilisateur INT NOT NULL,
    id_ressource INT NOT NULL,
    note DECIMAL(2,1) NOT NULL,
    critique TEXT,
    date_evaluation DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    UNIQUE KEY unique_evaluation (id_utilisateur, id_ressource),
    CONSTRAINT fk_eval_utilisateur FOREIGN KEY (id_utilisateur)
        REFERENCES utilisateur(id_utilisateur)
        ON DELETE CASCADE,
    CONSTRAINT fk_eval_ressource FOREIGN KEY (id_ressource)
        REFERENCES ressource(id_ressource)
        ON DELETE CASCADE,
    INDEX idx_ressource (id_ressource),
    INDEX idx_utilisateur (id_utilisateur),
    INDEX idx_note (note)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

SHOW TABLES;

```