Une image contenant conception

Description générée automatiquement avec une confiance moyenne

Projet BDR

Pierric Ripoll, Lisa Gorgerat, Alex Berberat

**Table des matières :**

[1. Introduction 3](#_Toc188813644)

[Objectifs du projet 3](#_Toc188813645)

[Données des médias 3](#_Toc188813646)

[Fonctionnalités principales 3](#_Toc188813647)

[2. Modèle EA 4](#_Toc188813648)

[Description : 4](#_Toc188813649)

[3. Modèle relationnel 5](#_Toc188813650)

[4. Manuel utilisateur 6](#_Toc188813651)

[Général 6](#_Toc188813652)

[Pages spécifiques 7](#_Toc188813653)

[5. Bugs connus 9](#_Toc188813654)

[6. Conclusion 9](#_Toc188813655)

[7. Annexes 10](#_Toc188813656)

[Guide d’installation/de déploiement 10](#_Toc188813657)

[Prérequis généraux 10](#_Toc188813658)

[Recommandation : 10](#_Toc188813659)

[En tant qu'utilisateur 10](#_Toc188813660)

[En tant que développeur 10](#_Toc188813661)

[Configuration à l'aide de GitHub 10](#_Toc188813662)

[Construire le code source de l'application 10](#_Toc188813663)

[Construire le paquetage de l'application 10](#_Toc188813664)

[Utilisation locale 11](#_Toc188813665)

[Etapes de déploiement 11](#_Toc188813666)

[Avec Docker 11](#_Toc188813667)

[Sans Docker 11](#_Toc188813668)

# Introduction

Ce projet a pour objectif de concevoir et développer une **application web interactive** permettant aux utilisateurs de créer et de maintenir une liste personnelle de médias de différents types.

## Objectifs du projet

L'application offrira aux utilisateurs la possibilité de :

* **Créer une liste personnalisée de médias** de types variés tels que les films, les livres, les bandes dessinées, les jeux vidéo et les séries.
* **Annoter les médias** en attribuant des notes et en ajoutant des commentaires.

## Données des médias

Chaque média aura des données générales et spécifiques :

* **Caractéristiques générales** : nom, date de sortie, genres, et créateurs
* **Caractéristiques spécifiques** selon le type de média :
  + Films : durée.
  + Livres : format (roman, recueil, essai, etc.).
  + Bandes dessinées : pas de spécificités supplémentaires.
  + Jeux vidéo : type de jeu (FPS, RPG, etc.).
  + Séries : nombre d'épisodes et de saisons.

## Fonctionnalités principales

L'application est dotée de fonctionnalités clés telles que :

* **Création de comptes utilisateurs** : chaque utilisateur pourra s'inscrire et gérer ses listes personnalisées.
* **Ajout et gestion des médias** : possibilité de rechercher des médias existants ou d’ajouter de nouveaux éléments à la base de données.
* **Commentaires personnalisés** : les utilisateurs pourront rédiger des avis et les associer à leurs médias favoris.

# Modèle EA

Une image contenant diagramme, Plan, Dessin technique, schématique

Description générée automatiquement

## Description :

**Principaux concepts :**

1. **Média** :
   * Une entité centrale représentant les attributs communs de formes de médias variés (livres, films, séries, jeux vidéo, etc.).
   * Les médias sont organisés en catégories distinctes (papier ou numérique) avec des spécialisations (par exemple, livre, film, série, etc.), ce qui permet un tri plus précis.
2. **Utilisateur** :
   * Représente les personnes utilisant le système.
   * Les utilisateurs peuvent créer des listes de médias (comme "à voir", "favoris", etc.) et ajouter des commentaires sur les médias qu'ils suivent.
3. **Genre** :
   * Permet de classifier les médias par thème ou catégorie (comme science-fiction, comédie, etc.).
4. **Créateur** :
   * Définit les auteurs ou groupes responsables de la création des médias.
   * Les créateurs peuvent être des personnes individuelles ou des groupes avec les attributs commun dans créateurs.
5. **Commentaire** :
   * Permet aux utilisateurs de donner leur avis sur un média. Les commentaires incluent une note, une date et un texte.
6. **Liste** :
   * Représente des collections personnelles de médias créées par les utilisateurs, comme une liste de visionnage ou de lecture.

**Relations principales :**

* Les médias appartiennent à des genres.
* Les utilisateurs peuvent commenter des médias, mais seulement après les avoir ajoutés à une liste.
* Les médias sont associés à des créateurs qui les ont produits.
* Les listes permettent aux utilisateurs de regrouper et d’organiser les médias selon leurs préférences.

# Modèle relationnel

Createur(id, nom)

Groupe(id)  
Groupe.id references Createur.id

Personne(id, prenom)  
Personne.id references Createur.id

Genre(nom)

Media(id, nom, dateSortie, description)

Media\_Createur(idMedia, idCreateur) Media\_Createur.idMedia references Media.id Media\_Createur.idCreateur references Createur.id

Media\_Genre(idMedia, nomGenre) Media\_Genre.idMedia references Media.id Media\_Genre.nomGenre references Genre.nom

Papier(id)  
Papier.id references Media.id

Livre(id, nbPages)  
Livre.id references Papier.id

BD(id, couleur)  
BD.id references Papier.id

Numerique(id)  
Numerique.id references Media.id

Film(id, duree)  
Film.id references Numerique.id

Serie(id, nbSaison)  
Serie.id references Numerique.id

Type(nom)

JeuVideo(id)  
JeuVideo.id references Numerique.id

JeuVideo\_Type(jeuVideoId, typeNom) JeuVideo\_Type.jeuVideoId references JeuVideo.id JeuVideo\_Type.typeNom references Type.nom

Utilisateur(pseudo, motDePasse)

Liste(pseudo, nom, dateCreation, mediaId)  
Liste.pseudo references Utilisateur.pseudo Liste.mediaId references Media.id

Media\_Liste(idMedia, pseudoListe, nomListe) Media\_Liste.idMedia references Media.id (Media\_Liste.pseudoListe, Media\_Liste) references (Liste.pseudo, Liste.nom)

Commentaire(pseudo, id, date, note, text)  
Commentaire.pseudo references Utilisateur.pseudo  
Commentaire.id references Media.id

# Manuel utilisateur

## Général

Lors de l’arrivée sur le site, nous pouvons faire une recherche ou regarder les derniers médias ajouter à la base de données.

Une image contenant texte, capture d’écran, logiciel, affichage

Description générée automatiquement

Nous pouvons regarder les médias dans la base de données avec la page Explore et ajouter un média. Grâce à la page Insert.



Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquementUne image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

Pour plus de fonctionnalités, comme voir ces listes personnalisées et ajouter des commentaires, nous devons nous connecter ou, si nous n’avons pas de compte, en créer un.

Une image contenant logiciel, Logiciel multimédia, capture d’écran

Description générée automatiquement

## Pages spécifiques

Les pages de login et de création de comptes sont basiques. Il suffit de rentrer son pseudo existant/voulu et son mot de passe.

Une image contenant texte, capture d’écran, Police, conception

Description générée automatiquementUne image contenant texte, capture d’écran, Police, logo

Description générée automatiquement

Pour l’insertion, il faut remplir le formulaire. Pour plusieurs sections, comme les genres, on peut ajouter des éléments existants, autrement, il y a une zone pour ajouter ce que nous voulons.

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

Dans la page des listes, la zone out en haut de la page permet la création d’une nouvelle liste.

Une image contenant capture d’écran, texte, ligne, Police

Description générée automatiquement

Dans la page d’un média, la zone en haut à gauche permet d’ajouter un média à une liste et la zone après les informations, mais avant les commentaires, permet d’ajouter un commentaire.

Une image contenant texte, capture d’écran, diagramme, nombre

Description générée automatiquement

# Bugs connus

1. La fonction de recherche ne fonctionne pas
2. La note moyenne d’un média s’affiche mal

# Conclusion

Comme ce projet était en commun avec DAI, nous avons eu des complications qui n’étaient pas liées avec la partie connectée avec la base de données.

Concernant la partie purement BDR, nous n’avons pas eu le temps de faire certaines fonctionnalités, comme la modification ou suppression de médias depuis le site ou encore des recommandations, mais nous avons réussi à implémenter la majorité des fonctionnalités qui étaient prévues à la base.

Nous avons eu quelques défis avec le SQL, comme pour faire l’insertion de médias, due au fait qu’il y a beaucoup de tables et un nombre variable d’attributs ou encore la requête pour la recherche. Généralement, notre problème était le cas où le nombre de paramètres à mettre dans la requête était variable.

Nous avons pu apprendre à faire et utiliser des requêtes compliquées utilisant, par exemple, des fonctions et triggers. Ce qui a été intéressant fut trouver comment créer du contenu au hasard pour la base donnée.

Si nous voulions continuer à développer cette application, pour pourrions ajouter les fonctionnalités suivantes :

* Modification des médias
* Suppression de médias
* Page de profile
  + Possibilités de modifications liées
* Faire des recommandations de médias similaires

# Annexes

## Guide d’installation/de déploiement

#### Prérequis généraux

* Docker
* Docker Compose

Pour utiliser l'image Docker, vous devez d'abord vous assurer que Docker est installé sur votre machine. Si ce n'est pas le cas, veuillez-vous rendre sur le site officiel ([Dockerdocs](https://docs.docker.com/get-started/get-docker/)) et suivre les instructions pour la version dont vous avez besoin.

##### Recommandation :

Utilisez Docker et Docker compose au lieu d'exécuter l'application en mode natif, c'est beaucoup plus facile.

#### En tant qu'utilisateur

Pour commencer, vous aurez besoin de notre paquetage d'application que vous pouvez trouver [ici](https://github.com/users/LisaGorgerat/packages/container/package/webserver).

Vous pouvez également utiliser cette commande dans votre terminal :

docker pull ghcr.io/lisagorgerat/webserver:latest

Vous pouvez ensuite passer à la section déploiement.

#### En tant que développeur

Prérequis supplémentaires :

* JDK
* Maven (optionnel, un wrapper maven est fourni avec le projet)

##### Configuration à l'aide de GitHub

Allez sur le dépôt sur GitHub et choisissez votre option de clonage préférée.

Ouvrez le terminal dans le dossier où vous voulez cloner le dépôt.

Clonez le dépôt.

git clone https://github.com/Nyaaw/ProjectBDR.git

Changez l'url de git remote pour éviter les pushs accidentels vers le projet de base.

git remote set-url origin <github\_username/repo\_name>

git remote -v

##### Construire le code source de l'application

Pour construire l'application, vous pouvez utiliser les commandes suivantes dans votre terminal.

./mvnw dependency:go-offline

./mvnw clean package

##### Construire le paquetage de l'application

Si vous deviez modifier notre application, vous devrez la compiler à nouveau et la publier sur votre compte.

Pour construire l'application :

Allez dans le dossier de l'application et utilisez la commande :

docker build -t webserver .

Pour publier le conteneur sur GitHub :

* Vous devez d'abord vous connecter à votre compte :
* docker login ghcr.io -u <username>

Vous devez package l'image :

docker tag webserver ghcr.io/<username>/webserver:latest

Vous publiez le conteneur sur votre compte :

docker push ghcr.io/<username>/webserver

#### Utilisation locale

Pour exécuter le projet localement, vous devez disposer d'une base de données Postgre avec les paramètres suivants (si vous modifiez l'un des éléments suivants, n'oubliez pas de modifier le fichier src/main/java/ch/heigvd/dai/Main.java).

* Nom de la base de données : mediatheque

Nous avons des scripts sql pour créer le contenu de la base de données et pour lui donner des données de base, si vous voulez la remplir. Les scripts se trouvent dans docs/BDR-phases dans les phases 3 et 4. L'ordre d'exécution est DDL.sql, DML.sql et base\_struct\_and\_data.sql.

##### Etapes de déploiement

###### Avec Docker

Pour exécuter le projet avec Docker, après avoir empaqueté et construit le projet avec Docker, utilisez la commande suivante :

docker compose up

Si vous l'exécutez en local, vous pouvez y accéder en allant à l'adresse localhost:8080 dans votre navigateur.

###### Sans Docker

Pour exécuter le projet sans Docker, après l'avoir empaqueté, exécutez la commande suivante :

java -jar target/webserver.jar

Pour voir le résultat, allez à l'adresse localhost:8080 dans votre navigateur.