

Systèmes d'Information – projet 2025/26

Master 1^{ère} année ILIADE & THIL-A – UBO

Ce projet, à réaliser par groupe de 3 étudiants, compte pour la note de contrôle continu de l'UE Systèmes d'Information ($\frac{1}{3}$ de la note de l'UE).

Il s'agit d'implémenter une application Web de gestion de vidéos à la demande en définissant plusieurs services implémentés par différentes technologies (Servlet et Spring) et s'appuyant sur plusieurs serveurs de bases de données (SQL et MongoDB).

1. Description de l'application

L'application à réaliser sert à gérer des réservations de films par un système de VOD (*Video On Demand*). On ne s'intéresse qu'au système de gestion des réservations sans s'occuper de la partie vidéo.

Pour accéder à l'application, un utilisateur doit créer un compte en renseignant son pseudo (qui servira d'identifiant de connexion), ses informations personnelles (nom, prénom, âge et adresse) et un mot de passe. Le pseudo doit être unique parmi tous les utilisateurs.

Du point de vue de l'utilisateur, l'application permet de consulter la liste des films, la liste des acteurs et actrices, la liste des réalisateurs et réalisatrices ainsi que les films d'un certain réalisateur, d'un certain acteur ou d'un certain genre. Un film est défini par un titre (unique entre tous les films), une année de réalisation, un réalisateur, un ensemble d'acteurs, des genres ainsi qu'un âge minimum pour pouvoir regarder le film. On peut ajouter une affiche à un film (fichier au format image). Un film est mis en location à un certain prix.

L'utilisateur, une fois connecté avec son pseudo et son mot de passe, peut réserver des films. Il peut au maximum avoir 3 films en réservation en même temps. L'utilisateur peut terminer la location d'un film quand il le souhaite. Il ne peut réserver un film que s'il est suffisamment âgé par rapport à l'âge minimal spécifié pour le film. L'utilisateur connecté peut récupérer la liste des films qu'il a actuellement en location.

Pour finaliser une réservation, l'utilisateur devra la payer. Pour cela, on mettra en place une API de pseudo-paiement par carte bancaire qui simulera le paiement (avec des échecs dans certains cas).

Un utilisateur peut ajouter une évaluation à un film. Une évaluation consiste en une note qui est un entier entre 0 et 5 et un commentaire textuel facultatif. Un utilisateur ne peut faire un commentaire sur un film que s'il est connecté et qu'il a réservé le film (actuellement en réservation ou réservé dans le passé). Sans avoir besoin d'être connecté, on peut consulter toutes les évaluations associées à un film (on verra alors pour chaque évaluation, le pseudo de

l'utilisateur, sa note et son éventuel commentaire). Il est possible également de récupérer la moyenne des évaluations d'un certain film.

Du point de vue de l'administrateur, il y a des fonctionnalités pour créer de nouveaux films et de nouveaux artistes (qui seront acteurs ou réalisateurs) en initialisant tous les champs requis.

Un administrateur peut ouvrir un film à la location et le fermer à la location. Par défaut, un film est créé en étant fermé à la location. Un film fermé à la location apparaît tout de même dans la liste de tous les films et on peut en consulter les informations et les évaluations. Si un film était en cours de location par des utilisateurs quand il a été fermé, il est toujours considéré en location pour ces utilisateurs tant qu'ils n'en ont pas terminé la location. Lors de l'ouverture d'un film à la location, l'administrateur décide du prix. Ce prix peut être modifié pendant la période de location pour faire des promotions par exemple.

Pour peupler la base de données des films, une fonctionnalité permettra de récupérer des données en interrogeant des API de bases extérieures sur le Web (à vous de trouver lesquelles).

2. Contraintes techniques

La ou les bases de données stockant les informations sur les films et les réservations sont des bases SQL. L'accès aux données se fera via des entités JPA. Les affiches des films et les évaluations associées aux films sont stockées dans une base MongoDB.

L'interface client Web est à développer en utilisant le framework Vue.js. Elle permet aux utilisateurs et à l'administrateur d'utiliser tous les services de l'application.

Plusieurs API REST seront développées pour implémenter les fonctionnalités, elles seront définies pendant les TP. Les API d'accès aux films et de réservations seront développées en Spring. Les API d'accès aux évaluations et aux affiches des films seront développées avec des Servlet.

Pour partager le code au sein de votre trinôme, vous utiliserez Git.

Au sein d'un même groupe de TP (TP1, TP2 ou TP3), chaque trinôme développe un certain nombre de services REST (cela sera défini pendant les TP) et l'intégralité de la partie client. L'idée est que pour former l'application complète, les services d'un trinôme s'appuient sur les services implémentés par d'autres trinômes du même groupe de TP. Un serveur Docker sera mis à disposition pour déployer les services REST de chaque trinôme.

3. Evaluation

Evaluation sur machine

Dates d'évaluation du fonctionnement des projets en fonction du groupe de TP (voir emploi du temps ADE - susceptible d'être modifié) :

- TP1 : Jeudi 5 Mars à 13h30
- TP2 : Jeudi 5 Mars à 15h45
- TP3 : Vendredi 6 Mars à 10h15

Dépôt sur Moodle

Pour le Vendredi 6 Mars à 19h au plus tard, vous déposerez sur le Moodle de l'UE, le code complet de votre application (gestion des données, code serveur et client HTML/JS).

Vous joindrez dans l'archive, un fichier PDF dans lequel vous détaillerez votre schéma de BDD, la structure des POJO ainsi que la liste des méthodes d'accès aux données. L'idée est qu'il y a une couche données qui sera utilisée par les Servlet et sera bien découplée de l'implémentation des API de la partie métier Web. Vous détaillerez également dans le document les API REST que vous avez définies en précisant pour chaque API avec quelle technologie elle a été implémentée.

Le dépôt pour la partie Spring sera précisé ultérieurement.