

Travail pratique #3 : Conception du serveur d'application et du client

Cours	8TRD157 – Base de données avancées
Étudiant	Ann-Laurie Lapointe LAPA07540005

Rapport	(40 pts)
Manuel utilisateur	/ 10
Stratégie de test	/ 5
Patrons de conception	/ 10
Validation des contraintes et de l'accès	/ 5
Code de mappage et justification	/ 5
Question 1	/ 5
Code source	(35 pts)
Code Hibernate	/ 5
Code d'application Java	/ 30
Qualité du code	
Fonctionnalité	(25 pts)
Correction interactive	/ 25
Total	/ 100

1 Manuel utilisateur

Cas 2 : Connexion au système

Le premier écran au démarrage de l'application est l'interface de connexion utilisateur.



The screenshot shows a light blue background with the title "Connexion utilisateur" in bold black text. Below the title, there are two input fields: "Adresse courriel:" with the value "jeanbouchard@gmail.com" and "Mot de passe:" with the value "*****". A "Se connecter" button is positioned below the password field.

Il faut entrer l'adresse courriel et le mot de passe correspondants à l'un des utilisateurs dans la base de données puis appuyer sur **Se connecter**. Des exemples d'utilisateurs valides insérés avec data.sql :

Adresse courriel : jeanbouchard@gmail.com
Mot de passe : mdp1234

Adresse courriel : nancyg@gmail.fr
Mot de passe : biscuit564

Si l'un des champs est vide ou si les identifiants sont invalides, un message d'erreur s'affiche à l'utilisateur. Sinon, l'utilisateur passe à l'interface de location de film.



The screenshot shows a light green background with the title "Recherche de film" in bold black text. On the left, there are several input fields for search criteria: "Titre du film:", "Année min de sortie:", "Année max de sortie:", "Langue originale du film:", "Nom du réalisateur:", "Pays de production", "Genre(s)", and "Acteur(s)". A "Rechercher" button is at the bottom. On the right, there is a large white box titled "Liste de films correspondants".

Cas 3 : Consultation de films

Recherche de film

Titre du film:

Année min de sortie:

2000

Année max de sortie:

2015

Langue originale du film:

anglais

Nom du réalisateur:

Pays de production

Genre(s)

Acteur(s)

canada

Science-fiction

drame

Rechercher

Liste de films correspondants

Inception (2010)
Interstellaire (2014)

Dans l'interface de consultation de films, l'utilisateur peut entrer n'importe quelle combinaison de critères de recherche disponibles et appuyer sur **Rechercher** pour voir apparaître la liste de films qui correspondent aux critères sous le format <Titre du film (Année de parution)>. Les majuscules et les minuscules n'importent pas et les champs pour le titre du film, le nom du réalisateur et les noms des acteurs acceptent une chaîne de caractères se trouvant n'importe où dans le nom. Les autres champs doivent toutefois avoir le mot complet. Les champs d'années doivent avoir une valeur numérique. Si aucun champ n'est rempli, un message d'erreur est affiché à l'utilisateur.

Si aucun film correspondant aux critères n'est trouvé, un message est affiché à l'utilisateur. Sinon, l'utilisateur peut sélectionner un des films dans la liste.

Recherche de film

Titre du film:

Année min de sortie:

2000

Année max de sortie:

2015

Langue originale du film:

anglais

Nom du réalisateur:

Pays de production

Genre(s)

Acteur(s)

canada

Science-fiction

drame

Rechercher

Liste de films correspondants

Inception (2010)
Interstellaire (2014)

Interstellaire (2014)

Langue originale: Anglais

Durée: 169 minutes

Réalisateur: Christopher Nolan

Dans un proche futur, la Terre est devenue hostile pour l'homme. Les tempêtes de sable sont fréquentes et il n'y a plus que le maïs qui peut être cultivé, en raison d'un sol trop aride. Cooper est un pilote, recyclé en agriculteur, qui vit avec son fils et sa fille dans la ferme familiale.

Acteurs - Personnages

Matthew McConaughey - Joseph Cooper
Anne Hathaway - Amelia Brand

Pays de production

États-Unis
Canada

Scénaristes

Christopher Nolan

Genres

Drame
Science-fiction

Louer le film

Lorsque l'utilisateur clique sur un des films dans la liste, la fiche du film apparaît avec toutes les informations sur le film. L'utilisateur a alors l'option de cliquer sur le nom du réalisateur ou alors le nom d'un des acteurs pour faire apparaître la fiche de cette personne.

Interstellaire (2014)

Langue originale: Anglais
Durée: 169 minutes
Réalisateur: Christopher Nolan

Dans un proche futur, la Terre est devenue hostile pour l'homme. Les tempêtes de sable sont fréquentes et il n'y a plus que le maïs qui peut être cultivé, en raison d'un sol trop aride. Cooper est un pilote, recyclé en agriculteur, qui vit avec son fils et sa fille dans la ferme familiale.

Acteurs - Personnages
Matthew McConaughey - Joseph Cooper
Anne Hathaway - Amelia Brand

Scénaristes
Christopher Nolan

Pays de production
États-Unis
Canada

Genres
Drame
Science-fiction

Louer le film

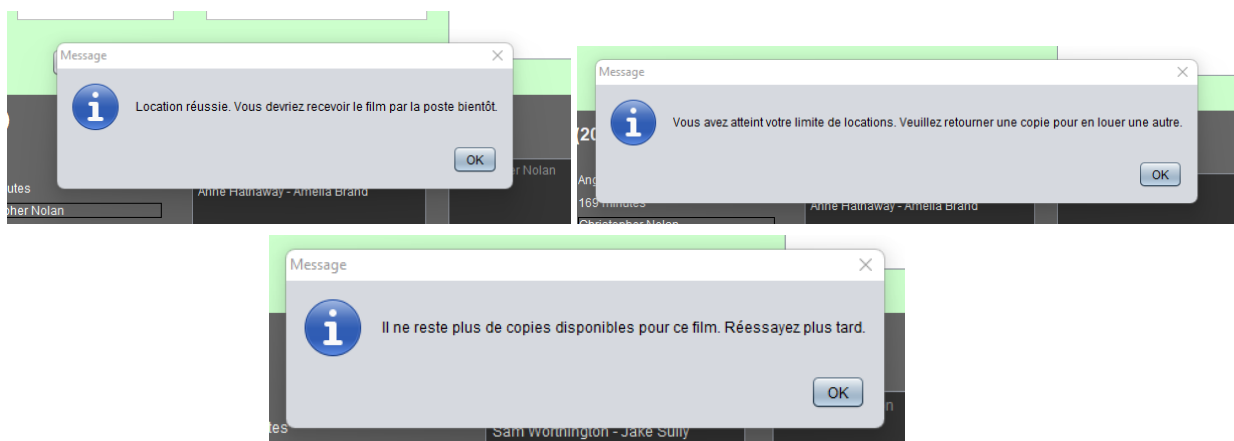
Matthew McConaughey
Acteur, réalisateur, scénariste et producteur de cinéma américain.

Date de naissance: 1969-11-04
Lieu de naissance: Uvalde

L'utilisateur peut à tout moment sélectionner une autre personne, un autre film ou effectuer une autre recherche.

Cas 4 : Location de films

Si l'utilisateur décide de louer le film qu'il est en train de consulter, il doit appuyer sur **Louer le film**. Si aucune copie du film n'est disponible ou si la limite de locations de l'utilisateur est atteinte, un message d'erreur est affiché au client. Sinon, un message de succès lui est affiché, une nouvelle location active est créée à son actif et l'inventaire des copies est mis à jour.



2 Stratégie de test

À chaque nouvelle fonctionnalité implémentée dans mon application, je démarrais l'application pour la tester en essayant toutes les possibilités. Par exemple, au niveau de la consultation de films, j'ai effectué de nombreux tests avec toutes les combinaisons de critères possibles pour m'assurer que peu importe quels champs sont remplis, la recherche se fait correctement. J'ai régulièrement testé le processus du début à la fin, de la connexion à la location de films, en me mettant dans la peau d'un utilisateur et en essayant toutes les séquences d'actions possibles. J'ai aussi essayé en étant connectée avec des utilisateurs différents pour m'assurer que la validation au niveau du maximum de locations était bien fonctionnelle.

3 Patrons de conception employés

J'ai employé le patron de courtier BD DAO, ou *Data Access Object*, dans mon application. Ce patron a pour rôle d'isoler l'application de la couche de persistance, soit la base de données relationnelle. Ainsi, la complexité de l'accès à la base de données est cachée de l'application et toutes les interactions avec la base de données sont isolées. Dans mon application, j'ai fait un seul courtier, *CourtierDaoImpl*, dans lequel toutes les opérations nécessitant un accès à la base de données sont isolées dans leurs propres méthodes; ainsi, aucune interaction avec la base de données n'est directement faite dans l'application, seules les méthodes du courtier sont appelées.

4 Validation des contraintes

La validation des contraintes se fait au niveau de l'application. Par exemple, lorsque l'utilisateur appuie sur Louer le film, l'application va chercher le nombre de locations maximal de son forfait ainsi que le nombre de locations à son actif et vérifie si le nombre de locations actives est inférieur au nombre de locations maximal. Si ce n'est pas le cas, un message est présenté à l'utilisateur dans l'interface et l'application n'essaie pas de faire la location.

5 Validation de l'accès et sécurité du système

L'application contrôle l'accès. Elle ouvre sur l'écran de connexion et tant que les bons identifiants ne sont pas entrés sur l'écran de connexion, elle n'affiche pas l'interface de location de film. Le mot de passe de l'utilisateur est sécurisé de deux façons. D'abord en surface sur l'écran de connexion, où le mot de passe entré est masqué par des ****. Ensuite du côté de la base de données; le mot de passe y est encrypté et pour comparer le mot de passe entré avec celui dans la base de données, le mot de passe entré sur l'écran de connexion est également encrypté dans l'application puis c'est cette version encryptée qui est comparée avec la base de données. Le mot de passe dans la base de données n'est donc jamais compromis.

6 Modifications au schéma

Peu de changements ont été apportés à mes schémas. Au dernier laboratoire, j'ai fait une table séparée pour les adresses des utilisateurs afin de rendre la table Utilisateur un peu moins chargée. Dans ce laboratoire, le seul changement apporté a été d'ajouter une colonne idActeur dans la table Acteurfilm puisque, étant donné qu'il s'agit d'une table associative et qu'elle a une colonne supplémentaire pour le nom du personnage, Hibernate avait besoin d'une colonne de clé primaire. Également, ceci n'est pas au niveau de mes schémas mais j'ai apporté des modifications à mon fichier cretab.sql pour que les mots de passe des utilisateurs soient encryptés dans la base de données, chose que j'ai oublié de faire au dernier laboratoire. J'ai aussi rajouté quelques données dans mon fichier data.sql (pays, genres, scénaristes) pour avoir un peu plus d'éléments dans mon application. Ma remise inclue ces deux fichiers modifiés.

7 Difficultés rencontrées avec Hibernate

Comme dit plus haut, j'ai eu des problématiques avec ma table Acteurfilm qui est une table associative contenant une colonne supplémentaire. En raison de cette colonne supplémentaire, Hibernate ne voulait pas accepter cette table tant que je ne lui ajoutais pas une clé primaire, ce que j'ai donc dû faire. Également, Hibernate a mappé tous les ID clés primaires de mes tables avec un générateur de classe *assigned*, ce qui voulait dire que je devais manuellement insérer les ID lors d'insertions. J'ai donc dû les changer pour les assigner à leurs séquences Oracle et ainsi qu'ils s'incrémentent automatiquement. À part ça, je n'ai pas vraiment eu de difficultés. Hibernate a fait un bon mapping de ma base de données.

8 Planification des tâches

J'ai travaillé seule, donc j'ai effectué 100% du travail.

9 Question théorique 1

L'utilisation d'un framework ORM a plusieurs avantages; notamment, ça permet d'éviter la répétition de code SQL et ça rend le développement plus orienté objet, ce qui fait que le code est plus facile à maintenir et à réutiliser.

Toutefois, il y a des cas où il est préférable de ne pas utiliser de framework ORM, notamment si notre application nécessite beaucoup de requêtes SQL complexes. Les framework ORM excellent avec les requêtes SQL simples mais ont de la difficulté lorsqu'elles augmentent en complexité. Ils sont aussi à éviter dans le cas où la performance et la vitesse sont de mise, puisqu'ils ne sont pas très rapides. Également, dans le cas où la sécurité des données est extrêmement importante, il peut être préférable d'y aller avec des procédures stockées puisqu'avec un ORM, n'importe quoi peut être modifié dans la base de données.