

Travail Pratique #3:**Conception du Serveur d'Application et du Client**

Michael Munger

Université du Québec à Chicoutimi

Département d'Informatique et de Mathématiques

TRD157 : Bases de Données Avancées

Jimmy G. Nault

29 Novembre 2021

Rapport	
Manuel Utilisateur	/10
Strategie de Test	/5
Patrons de Conception	/10
Validation des Contraintes et de l'Acces	/5
Code de Mappage et Justification	/5
Question 1	/5
Code Source	
Code Hibernate	/5
Code d'application Java	/30
Qualite du code	
Fonctionnalité	
Correction Interactive	/2
Total	/ 100

Travail Pratique #3: Conception du Serveur d'Application et du Client

1. Manuel utilisateur

Cas 2 : connexion d'un utilisateur au système

Pour la connexion, un email client et un mot de passe doivent être utilisés. Si les identifiants ne fonctionnent pas, aucune autre fonction ne sera disponible et un message d'erreur apparaîtra. Les identifiants ont la forme client#@domain.com pour le email et Client123# pour le mot de passe où # est un chiffre entre 1 et 5 inclusivement.

Cas 3 : consultation interactive des films

Après une connexion réussie, les critères de recherche sont disponibles. Un ou plusieurs critères peuvent être utilisés, mais ils sont utilisés par la recherche indépendamment, sauf pour les années. On pourrait donc chercher pour un film et un réalisateur et avoir un film répondant au critère du titre et un autre répondant au critère du réalisateur. Pour ce cas, les critères prennent les formes suivantes, où # est dans {1,2,3,4,5} :

- Titre : Film#
- Année Min ou Max : 190#
- Pays : {Canada, France, Mongolie}
- Langue : {Français, Anglais, Espagnol, Arabe, Japonnais, Russe}
- Genre : {Action, Romance, Horreur, Comedy, Drame}
- Réalisateur : PTP# pour le prénom et PTN# pour le nom
- Acteur : {ActP1, ActP2, ActP3} pour le prénom et {ActN1, ActN2, ActN3} pour le nom

Tous les critères sauf les années sont sensibles à la casse et cherchent le critère dans les mots, ce qui veut dire que le critère n'a pas besoin d'être écrit au long pour fonctionner.

Cas 4 : location de films

Lorsque la fiche d'un film est sélectionnée, cliquer sur le bouton Louer permet d'effectuer la location du film. Appuyer sur ce bouton vérifie la disponibilité du film et la capacité du client à le louer.

selon son forfait.

2. Stratégie de test

Pour vérifier l'application et les règles d'affaires, des test unitaires ont été effectués pour tester les connexion avec des caractères illégaux, tous les logins client disponibles et des variations inexacts des logins. Pour ce qui est des critères de recherches, ils ont tous été testé unitairement en variant l'ajout de caractères illégaux et en utilisant des lettres au niveau des années. Finalement, les listes de toutes sortes ont été testé en vérifiant que chaque élément marche et que les boutons n'active pas lorsqu'il n'y a pas de sélection et qu'il est impossible d'ajouter un élément aux listes.

3. Patrons de conception employés

Les difficultés rencontrées avec hibernate ont réduit grandement ma capacité à refactoriser mon programme. Le Lazy Loading a été utilisé pour ne créer les objets seulement lorsque le besoin se présente et la SessionFactory de Hibernate a été utilisé pour obtenir la session. J'ai tendance à refactoriser mon code plusieurs fois d'habitude mais, avec mon nouveau-né, je n'ai pas eu le temps. Dans une version future, les queries passeraient par une façade qui vérifierait l'input, afficherait les données et effectuerait les queries.

4. Validation des contraintes

Tous les triggers du TP2 sont toujours utilisés pour les vérifications au niveau de la base de donnée. Avant d'initier une query, les champs de paramètres sont vérifiés pour s'assurer qu'ils ne sont pas vides. Finalement, pour protéger les queries, une vérification par fonction est effectuée au niveau du client pour rejeter les paramètres comportant un caractère illégal (% , ' "). Dans une version future, une attention plus particulière serait portée sur la liste des caractères illégaux.

5. Validation de l'accès et sécurité du système

Sincèrement, je n'ai même pas pris le champs JpasswordField de Swing. La validation envoi les données en clair et la base de donnée vérifie que le client existe selon le email et mot de passe donné. Si l'un ou l'autre des identifiants sont incorrecte, une erreur est montrée. Dans une version future, le champs JpasswordField serait utilisé et les mots de passe sur la base de donnée serait encrypté. De plus, la librairie java.crypto serait utilisé pour chiffrer les communications.

6. Modifications au schéma

Je n'ai pas reçu la correction du TP2 et je n'ai pas eu le temps de le demander donc le schéma est le même que le TP2. J'ai cependant placé les données dans un nouvel utilisateur au lieu de system.

7. Difficultés rencontrées avec Hibernate

J'ai utilisé NetBeans et j'ai eu plusieurs difficultés, principalement avec la création et l'utilisation de la sessionFactory. Après avoir commencé avec la version 12.5, j'ai changé pour la version 10 avec Ubuntu car 12.5 avait des problèmes majeures avec l'activation d'Hibernate. La version 10 a permis de faire la base du travail mais, lors des tests, la SessionFactory ne fonctionnait pas. Ce fût un retour sur windows avec NetBeans 7.2 et la version 7 de jdk que j'ai réussis à compléter le travail. Mis à part les difficultés lié à l'IDE, tout a plutôt bien été. Le seul problème fût l'utilisation de la propriété entity au lieu de class dans les relation many-to-many qui créait un problème.

8. Planification des tâches

Comme pour les autres TP, il n'y a eu que Michael Munger dans cette équipe et donc j'ai fait tous le travail.

9. Question théorique 1

Comme avantages, on a la possibilité d'utiliser plusieurs fonctionnalités (polymorphisme, héritence) de l'orienté objet avec la base de données sans avoir à la modifier et un framework ORM permet de simplifier l'accès à la base de données. Cependant, lorsqu'il faut maintenir un haut niveau de sécurité sur les données, l'utilisation d'un framework ORM ne serait pas utile. Similairement, dans les cas tels que le traitement de transaction, un modèle objet pourrait simplement ne pas s'appliquer et donc un framework ORM ne serait pas utile.