

Hibernate Projet n°4

Structure de la base données :

On dispose d'une base de données composée de 4 tables. Une table advertisement qui représente les annonces du site. Une table user qui représente les utilisateurs du site. Une table offre qui représente les offres faites par des utilisateurs sur certaines annonces. Voir MCD et MLD de la base de données utilisée en pièce jointe au projet.

En ce qui concerne l'architecture de l'application, nous avons opté pour un model MVC pour nous faciliter la représentation de la circulation des données.

Nous avons donc un package View qui ne contient qu'une classe Window qui permet l'affichage des données dans la console.

Ensuite, nous avons un package Model qui correspond aux classes de données (User, Offer, Advertisement).

Enfin, dans le package Controller, nous avons mis une classe MainController qui met à jour les données du model affichées par la vue. Pour interroger/extraire les données de la base de données, nous avons mis en place des classes DAO (UserDAO, OfferDAO, AdvertisementDAO). Celles-ci sont aussi dans ce même package.

Explication de la structure :

On distingue **proposition** et **offre** bien que les 2 soit stockés dans la table Offre. Une **proposition** est faite par un utilisateur sur une annonce publiée par un autre utilisateur. On dit que l'utilisateur qui a publié cette annonce reçoit alors une **offre**. Un utilisateur a donc une liste de proposition qu'il fait sur des annonces et une liste d'offre qui représente les propositions des autres utilisateurs sur **ses** annonces.

On représente cela dans la base de données de façon simple. Une annonce est associée à un id unique « idAnnonce » et à un propriétaire « iduser ». Quand un utilisateur fait une proposition on crée une offre avec les « idOffre, idAnnonce, idAcheteur ». Avec l'idAnnonce on peut donc récupérer le propriétaire qui lui est associé avec l'iduser.

Fonctionnement de l'application

L'application est une vue console avec, comme moyen de navigation la possibilité d'entrer le numéro du menu auquel on veut accéder.

Un utilisateur peut se connecter ou non. Connecté ou pas un utilisateur peut simplement consulter des annonces soit par une fonction recherche, soit en choisissant une catégorie composée d'annonces à afficher. Il peut ensuite sélectionner une annonce pour afficher ses caractéristiques détaillées.

Un utilisateur peut également créer un compte ce qui lui permet par la suite de se connecter.

Un utilisateur connecté peut lui effectuer plusieurs actions en plus de parcourir les annonces :

- Consulter ses annonces, ajouter ou supprimer une annonce
- Faire des propositions sur un produit
- Consulter ses propositions, supprimer une proposition
- Consulter ses offres, accepter une offre reçue sur une de ses annonces, refuser une offre reçue sur une de ses annonces

Remarques utiles :

- Lors de l'appel des méthodes de la classe MainController « acceptOffer() » et « delUserAdvertisement », on obtient des erreurs même si « acceptOffer() » fonctionne partiellement. Celle-ci doit d'abord supprimer toutes les offres qui correspondent à la cette annonce puis elle doit supprimer l'annonce elle-même. Cependant, l'exécution de la méthode provoque l'erreur :

```
Hibernate: select advertise0_idAdvertisement as col_0_0_ from Advertisement advertise0_ where iduser=?
Hibernate: select offer0_idOffer as idoffer1_1_0_, offer0_idAdvertisement as idadvert3_1_0_, offer0_idBuyer as idbuyer4_1_0_, offer0_price_offer as price_of2_1_0_ from Offer offer0_ where offer0_idof
Hibernate: select offer0_idOffer as idoffer1_1_0_, offer0_idAdvertisement as idadvert3_1_0_, offer0_idBuyer as idbuyer4_1_0_, offer0_price_offer as price_of2_1_0_, advertise1_idAdvertisement as idAdv
janv. 22, 2021 7:24:54 PM org.hibernate.internal.ExceptionMapperStandardImpl mapManagedFlushFailure
ERROR: HH0000346: Error during managed flush [org.hibernate.PersistentObjectException: detached entity passed to persist: model.Offer]
Error while committing the transaction
```

Nous savons que cette erreur vient de la méthode « deleteAd(...) » de AdvertismentnDAO mais nous ne comprenons pas pourquoi. De la même manière la méthode « delUserAdvertisement » ne fonctionne pas.

- Un script SQL qui construit la base donnée avec 3 utilisateurs et 5 annonces déjà créés est jointe au projet. La base donnée doit s'appeler **bdhibernate**. Le fichier persistence.xml est en mode « create ».

- Quand un utilisateur refuse une offre, on la supprime de la base de données.
- Quand un utilisateur accepte une offre, on supprime toutes les autres propositions sur cette annonce, puis on retire l'annonce de la base de données.
- Lors de la création de compte et la connexion d'un utilisateur, le username et le password sont sensibles à la casse.
- Un package main a été ajouté afin d'intégrer la fonction main de l'application. Cette fonction ne contient que la création d'un objet Window qui lance l'application grâce à la fonction mainMenu().