TP Noté 2017

Vous devez créer une application **« myEpsi »** reposant sur un projet **JEE** permettant de gérer des annonces stockées dans une base de données HsqlDB. Une annonce sera définie par un identifiant unique numérique (compteur initialisé à 1) incrémenté automatiquement, un titre (texte de 140 caractères), une description (memo), un statut (temporaire, publié, vendu ou annulé), un nombre de vue, une date de création et de modification, un prix et le mail d’un acheteur. Une annonce qui n’est pas encore vendue ou annulée pourra être vue par tous les utilisateurs.

En arrivant sur la page d’accueil, un utilisateur pourra créer un compte avec un email (qui doit être l’identifiant unique), un mot de passe et une confirmation du mot de passe (pour vérifier qu’il est bien identique.

Un administrateur, créé au préalable pourra gérer les utilisateurs (création, modification, suppression) et les annonces (suppression). Il pourra également créer une annonce qui sera automatiquement publique.

La page d’accueil (une fois connecté) présentera la liste des annonces (titre + auteur + date de modification) disponibles (non temporaires, non vendues et non annulées) ainsi que les annonces de l’utilisateur connecté. Des icones devant le titre permettra de reconnaitre les annonces des autres utilisateurs ainsi que les annonces archivées. Il aura également accès à un bouton « créer ». Le bouton « Créer » permettra d’afficher le formulaire vide avec un bouton ajouter et un bouton annuler. En cliquant sur un ligne d’une annonce qu’il a écrit, l’utilisateur pourra afficher le détail, la modifier, la supprimer ou revenir à la page d’accueil. Si l’annonce est vendue, le mail de l’acheteur sera affiché et l’annonce ne pourra plus être modifiée.

Sur les annonces des autres utilisateurs, un bouton « Acheter » permettra de changer le statut de l’annonce. Le mail de l’utilisateur sera alors enregistrée dans l’annonce.

Une url permettra défficher directement une annonce en consultation avec l’id en paramètre. Par exemple : **http://localhost:8080/myEpsi/annonce?ID=1**. La réponse sera une page HTML de consultation.

Les actions seront implémentées en **Servlet** et les pages en **JSP**. Les pages devront être responsives.

Les logs devront être mis en place avec **log4J** et afficher une trace de niveau INFO à chaque début d’exécution d’une servlet. Chaque requête devra être logée en niveau DEBUG. Lors d’un test de connexion en erreur, le login testé sera affiché en niveau ERROR.

Au démarrage de l’application 2 logs afficheront en niveau ERROR le nombre d’annonces présentes en base et le nombre d’utilisateurs.

Un composant JMX permettra d’afficher dans la **Jconsole** le niveau en cours des logs, et permettra de le changer en cliquant sur un bouton debug, info ou error.

Un autre composant JMX permettra d’afficher le nombre d’annonces dans la base.

Des tests d’intégration devront être mis en place avec Junit dans le projets :

* Il n’est pas possible de créer 2 comptes avec le même email
* Un utilisateur ne peut pas supprimer une annonce qui ne lui appartient pas
* Un utilisateur ne peut pas voir les annonces archivés et privés des autres utilisateurs
* On ne peut pas supprimer le compte administrateur

# Environnement

Eclipse Luna

Tomcat 8.0.18

JDK 8.0.131

HsqlDB

# Livrables

Le war de déploiement

Les sources sur Git

La base de données

Les tests unitaires

|  |  |
| --- | --- |
| **Evaluation** | **Note** |
| Démarrage sans erreur du projet | 3 |
| Connexion avec un utilisateur existant | 2 |
| Affichage de la liste des annonces | 4 |
| Formulaire de création + insertion en base d’une annonce | 4 |
| Formulaire de modification/suppression d’une annonce | 2 |
| Changement en privé ou archivé d’une annonce | 2 |
| Création d’un utilisateur | 3 |
| Fonctionnement de l’url de consultation | 2 |
| Affichage des logs au démarrage | 1 |
| Gestion des logs avec les 3 niveaux | 2 |
| Supervision JMX des messages | 2 |
| Supervision JMX des niveaux de logs | 3 |
| Bon fonctionnement des tests unitaires | 2 |
| Fonctionnement en responsive | 2 |
| Déploiement avec le war | 2 |
| Publication sur Git | 2 |
| Organisation du projet : nommage, package, configuration | 2 |
| Total | 40 |