SR03 - Projet

1. Introduction
2. Technologies utilisées

JEE : servlets / JSP / javax.mail

SQL : Mysql / javax.sql

HTML / CSS / JS

1. Structure du Projet

Pour ce projet nous avons essayé de respecter au maximum le modèle MVC, nous avons donc :

* Modèle : Représentent toutes les classes contenant les informations. En particulier chaque table SQL possède une représentation, dans le package « beans ».
* Vue : Regroupe toutes les pages JSP servant à envoyer les pages HTML au navigateur
* Contrôleur : Permet l’accès aux données de la base de données mais aussi du serveur (sessions). Le contrôleur est composé des packages « controle » et « dao ».

1. Gestion des Utilisateurs

Les utilisateurs sont divisés en 2 groupes : les stagiaires et les administrateurs. Les administrateurs peuvent créer, supprimer et modifier n’importe quelle information de tous les utilisateurs. Les stagiaires ne peuvent quant à eux que modifier des informations sur leur compte.

La création d’un utilisateur va déclencher l’envoi d’un mail (en utilisant le SMTP de l’UTC) contenant les identifiants.

1. Sécurisation des pages

Pour ce projet nous avons décidé de ne pas utiliser de servlets de filtrage. Pour les remplacer nous avons utilisé une classe « ConnectionControle » permettant de sécuriser un peu plus les pages. En effet on utilise la session mais aussi des cookies « HTTPOnly » pour éviter tout vol de session.

Cette classe permet aussi de différencier les utilisateurs (stagiaire / administrateur) et ainsi de n’afficher que les informations disponibles pour chacun plutôt que de rediriger vers de nouvelles JSP.

Dans le cas où une page devait être complétement inaccessible on effectue une redirection par :

|  |
| --- |
| response.sendRedirect(" PAGE DE DESTINATION "); |

1. Gestion des parcours

Les parcours permettent le stockage du nombre de bonnes réponses d’un utilisateur, du temps qu’il a mis, mais aussi des réponses elles-mêmes. Il est ainsi possible d’arrêter un parcours pour le reprendre ultérieurement.

Les parcours sont mis à jour en temps réel grâce à des appels AJAX. On a ainsi un affichage en direct du nombre de bonnes réponses et du temps.

La gestion du temps est effectuée par le code suivant :

|  |
| --- |
| **long** startedTime = (**long**) request.getSession().getAttribute(  String.*valueOf*(questionnaire\_id));  **long** duree = p.duree.getTime()  + System.*currentTimeMillis*()  - startedTime;  p.duree.setTime(duree);  request.getSession().setAttribute(  String.*valueOf*(questionnaire\_id),  System.*currentTimeMillis*()); |

1. Conclusion