UTC – SR03

Rapport Projet

Questionnaire

Enseignant : Enrico Natalizio

Développés par : De Régibus Fabrice & Pelletier Thomas

1. Introduction

Le projet consistait à réaliser un site permettant la création d’un questionnaire et la réalisation de ceux-ci par des utilisateurs. Le projet était décomposé en plusieurs parties. Une première partie est la gestion des utilisateurs, composée de deux types d’utilisateur : l’administrateur, qui est capable de créer des comptes et de les gérer mais également de gérer les questionnaires ; et les stagiaires, qui sont capables de répondre aux questionnaires, de visualiser les scores et de le recommencer si besoin. La deuxième partie consistait à la gestion des questionnaires qui permettait à l’administrateur de modifier les questions, les réponses mais également l’ordre de ceux-ci. Cette partie devait permettre de visualiser les scores de l’ensemble des utilisateurs. La troisième partie consistait à parcourir les questionnaires, de visualiser les scores pendant le questionnaire et de visualiser le temps de réponse grâce à un chronomètre.

L’intégralité du projet n’a pas été fait et nous allons vous présenter ceux qui a été réalisé au cours de ce rapport.

1. Technologies utilisées

Pour ce projet, les technologies utilisées ont été les suivantes.

* JEE : servlets / JSP / javax.mail

Nous avons utilisé Java EE pour son usage des servlets et les JSP pour créer dynamiquement les pages. En complément, nous avons utilisé l’API JavaMail pour l’envoi de mail à la création de comtes.

* SQL : MySQL / javax.sql

Pour la gestion de nos données, nous avons utilisé la base de données MySQL car c’est cette base qui est utilisée au sein de l’UTC.

* HTML / CSS / JS

Nous avons utilisé les langages du web pour pouvoir structurer et styliser nos pages web.

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.

Voici la page d’accueil d’un administrateur, contenant la partir administrateur et la partie étudiant.



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 questionnaires

Les questionnaires sont gérés par les administrateurs. On peut les créer, les modifier ou les supprimer. La création d’un questionnaire nécessite de lui spécifier un nom et de lui choisir un sujet. Le sujet a été rajouté dans notre modèle de donnée mais aucune interface n’a été réalisé pour la modification de ceux-ci. Après avoir sélectionné le questionnaire, on peut modifier les questions et les réponses associées à ce questionnaire. On peut également modifier l’ordre des questions. Voici une page de modification d’une question.



1. Gestion des parcours

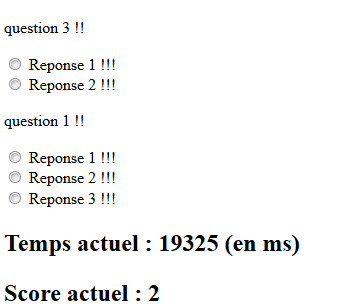
Les parcours permettent le stockage du nombre de bonnes réponses d’un utilisateur, du temps qu’il a mis, mais aussi des réponses elles-mêmes. Il est ainsi possible d’arrêter un parcours pour le reprendre ultérieurement. On peut également le réinitialiser pour le reprendre de zéro.

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*()); |

On peut visualiser ce chronomètre et le score réalisé à la fin de la page du questionnaire comme le montre l’image suivante.



1. Conclusion

Ce projet nous aura permis de mieux appréhender la programmation en Java EE avec la programmation des servlets qui nous permet de gérer chaque requête faite sur le serveur. Nous savons également maintenant comment récupérer les données sur une base de données MySQL. Même si tout n’a pas été réalisé, comme la gestion de la pagination et de la recherche, les principales fonctionnalités ont été réalisé.