Projet web

Java / JEE

Mickaël FARDILHA – Dorian COFFINET – Thibault GAUTHIER – Raphaël PILLIE

**

**

Sommaire

[Introduction 3](#_Toc385460438)

[I – Installation et environnement 3](#_Toc385460439)

[II – Conception et modélisation 3](#_Toc385460440)

[III – Déploiement sur une base MySQL locale 4](#_Toc385460441)

[IV – Déploiement sur une base MySQL distante préconfigurée 5](#_Toc385460442)

[V – Tests 5](#_Toc385460443)

[VI – Extensions 5](#_Toc385460444)

[VII – Cas utilisateur 5](#_Toc385460445)

[Conclusion 6](#_Toc385460446)

Introduction

L’objectif de ce rapport est de présenter le travail effectué suite à la réalisation du projet de java/jee.

C’est pourquoi, tout au long de celui-ci, nous allons détailler la réalisation de ce projet, en commençant par la conception et la mise en place de la structure, jusqu’à la réalisation et la finalisation.

I – Installation et environnement

En ce qui concerne l’environnement de développement, nous avons décidé d’utiliser un EDI, et plus précisément Eclipse, en version Kepler, auquel nous avons ajouté plusieurs outils et frameworks :

* Tomcat7, version 7.0.33
* JPA, version 2.1
* MySQL
* Struts², version 2.3.16

II – Conception et modélisation

Afin de mener à bien ce projet, nous avons décidé d’utiliser certains frameworks. Tout d’abord, nous avons mis en place la dernière version stable de struts 2, qui nous à permit de formater notre application avec le pattern MVC, de simplifier et d’assister nos développements.

Ensuite, nous avons mis en place une base MySQL pour gérer nos enregistrements.

Puis, pour lier notre application web à notre base de données, nous avons mise en place JPA, avec la persistance d’objets facilitant également nos développements.

Enfin, pour déployer et faire tourner notre application, nous avons décidé de mettre en place un serveur Tomcat.

Une fois la conception technique mise en place, nous nous sommes occupés de la conception fonctionnelle, et de la modélisation de la base de données.

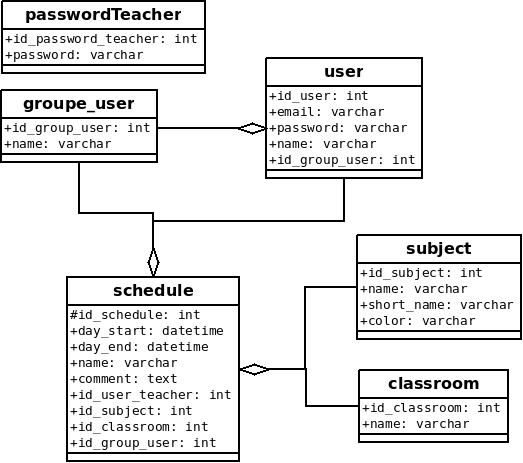
Ainsi, nous avons découpé les différents modules en différentes classe, toujours en respectant le pattern MVC.

Dans la partie contrôleur, nous avons créé une classe abstraite contenant des constantes et la variable de session, qui est la classe mère de toutes classes actions de la partie contrôleur.

Dans la partie modèle, Nous avons modélisé chacune de nos tables en base par des entités JPA. Puis, nous avons créé une classe avec des Templates, nommé abstractDAO dans laquelle nous avons créé les méthodes de sauvegarde, de suppression, de mise à jour, de récupération d'élément par leur identifiant et de récupération de tous les éléments.

Grace à cela, en faisant hériter nos classe DAO objets à celle-ci et en précisant dans cette classe fille le type de l'objet, il n'est pas nécessaire d'écrire pour chacun sous forme de persistance chacune des méthodes énoncées ci-dessus, puisqu'elles fonctionneront d'elle-même.

Enfin, nous avons modélisé la base de données de la façon suivante :



III – Déploiement sur une base MySQL locale

En prenant le livrable fournit, il est possible de déployer l’application sur une base locale.

Pour cela, il faut suivre la procédure suivante :

Tout d’abord, il faut passer les scripts SQL présent dans l’archive rendu, et dans le répertoire LocalDeployement, dans l’ordre suivant : initUser.sql, puis initTables.sql.

Ensuite, il ne reste qu’à déployer sous Tomcat7 le war présent dans le même répertoire.

IV – Déploiement sur une base MySQL distante préconfigurée

En prenant le livrable fournit, il est possible de déployer l’application sur une base MySQL distante, d’adresse IP : 5.48.29.15

Le déploiement est instantané, il suffit de déployer le war sur un serveur tomcat

V – Tests

Dans le cadre des tests réalisé pour rendre compte de la robustesse de notre application, nous avons mis en place des tests unitaire sur la partie modèle de notre application.

De plus, nous avons mis en place des tests d’intégration croisé à la fin de nos développement qui nous ont permis de trouver et de corriger certains bugs mineurs.

VI – Extensions

Malgré le peu de temps qui nous a été accordé pour mener ce projet à bien, nous avons mis en place une gestion complète de l’application en multilingue grâce à des fichiers de propriétés. Actuellement, l’application est traduite en Anglais et en français, en fonction de la langue du navigateur.

VII – Cas utilisateur

A l’adresse suivante, [*http://localhost:8080/ProjetEDT/*](http://localhost:8080/ProjetEDT/)

3 possibilités sont disponibles :

* Se connecter
* S’inscrire
* Afficher le calendrier

Pour se connecter il faut soit passer par l’inscription soit se connecter avec ces identifiants *professeurs : chhel@univ-angers.fr, mot de passe : chhel.*

Pour se connecter il faut renseigner :

* Email
* Mot de passe

Pour s’inscrire il faut renseigner :

* Prénom
* Nom
* Choisir son groupe. Si on choisit le groupe “Enseignant” il faut rentrer le champ “Mot de passe Enseignant sinon l’inscription échoue. Actuellement : passTeach
* Email
* Mot de passe
* Confirmation du mot de passe

Pour afficher le calendrier, il faut renseigner le formulaire :

* Possibilité de choisir le type de calendrier (Normal, Compact, Résumé)
* Choisir le groupe auquel est rattaché la calendrier
* La date de début
* La date de fin
* Valider vos choix.

En tant qu’enseignant il est possible d’accéder aux modules suivants :

* Utilisateurs (modification/suppression)
* Groupes (ajout/modification/suppression)
* Salles (ajout/modification/suppression)
* Horaires (ajout/modification/suppression)

Conclusion

Pour conclure ce rapport, il est important de souligner que ce projet nous a permis d’apprendre et de pouvoir mettre en place une solution de la conception jusqu’à la réalisation, en passant par différentes étapes, en autonomie total. Cela permet également de développer un esprit critique sur la technologie ainsi que d’explorer une autre vision de l’apprentissage que celle fournie par les enseignements universitaires.