Département Génie Informatique
Projets TP LO54

Deux parties sont à réaliser :

- La première partie est à réaliser avec les technologies suivantes : hibernate, servlet, jsp. Pour la présentation des pages, utilisez bootstrap, ne par chercher à utiliser d'autres framework js.
  La note portera uniquement sur le code et le respect des conventions de codage vu en cours et TD.
  Livrables / Deliverables :
  Vous devz rendre dans un . zip le code source de votre projet.
  La date de livraison sera précisée dans l'agenda Moodle.

- La deuxieme partie est la continuité de la partie 1, vous devez ajouter à votre projet une technologie complémentaire.
  La note portera :
  - *sur un document qui contiendra :*
    - *Un rapport d'experience (*vécu sur la technologie sur la technologie)
    - *Un tutoriel d'utilisation de la technologie*
  - *une présentation (soutenance de 15 mn) ou vous présenterez la technologie et l'integration que vous en avez faite dans votre projet.*

Par groupe de 2 ou 3 max.

Pour éditer votre code vous pouvez utiliser l'IDE de votre choix (Eclipse / Netbeans / Intellij /….)

La structure de votre projet doit respecter l'architecture SOA et Maven.

1) PARTIE 1 SUJET PRINCIPAL

Il s'agit de gérer l'offre de formation d'une école privée.

L'école édite un catalogue en ligne (HTML) listant toutes les formations disponibles et pour chaque formation, les dates des prochaines sessions prévues.

L'utilisateur doit être capable de :

- filtrer la liste des formations par un mot clé contenu dans le titre de la formation.
- filtrer la liste des formations par les sessions disponibles à une date donnée.
- filtrer la liste des formations en fonction du lieu de la session, ce lieu provenant d'une liste déroulante.

L'utilisateur doit pouvoir sélectionner une session et s'y pré-inscrire en indiquant ses coordonnées personnelles (Nom, Prénom, Adresse, Téléphone, Email).
Pour ne pas compliquer le modèle, on considérera qu'une même personne s'inscrivant à 2 sessions est alors présente 2 fois dans la base.

Il n'est pas demandé de créer une interface d'administration pour gérer les données de la base.

<u>On disposera donc des tables suivantes</u> :
La création est à votre charge, vous pouvez utiliser le SGBD que vous voulez (MySQL/POSTGRES/ORACLE ou JavaDB)

LOCATION
- ID → Number AUTO (PK)
- CITY → Char Not Null

COURSE
- CODE → Char Not Null (PK)
- TITLE → Char Not Null

COURSE_SESSION
- ID → Number AUTO (PK)
- START_DATE → Date Not Null
- END_DATE → Date Not Null
- COURSE_CODE → Char Not Null (FK)
- LOCATION_ID → Number Not Null (FK)

CLIENT
- ID → Number AUTO (PK)
- LASTNAME → Char Not Null
- FIRSTNAME → Char Not Null
- ADDRESS → Char Not Null
- PHONE → Char Not Null
- EMAIL → Char
- COURSE_SESSION_ID → Number Not Null (FK)

2) PARTIE 2 FONCTIONNALITES COMPLEMENTAIRE

Vous devez approfondire une des technologies suivante et ajouter le code correspondant au sujet 1 :

- JPA 2

Exploitez JPA2 avec Eclipse Link pour le requêtage.

- Spring Data

Exploitez Spring data pour le requêtage de la liste des cours.

- JUNIT

Réalisez les tests unitaires de vos services métiers avec JUNIT

- BIRT

Ajoutez une fonction permettant de générer un PDF obtenu grâce à BIRT sur la base d'un template personnalisé.
Ce PDF affichera la liste des inscrits par session

- JASPER REPORT

Ajoutez une fonction permettant de générer un PDF obtenu grâce à JASPER REPORT sur la base d'un template personnalisé.
Ce PDF affichera la liste des inscrits par session

- JAX-RS

Transformez les fonctions de recherche en WebServices REST.
Les résultats seront fournis sans rechargement de page via une requête JSON au WebService REST. Utilisez l'implémentation de JAX-RS de votre choix (sauf Jersey) pour réaliser les services.

- PrimeFaces

Réaliser la couche de contrôle à l'aide de l'implémentation JSF Primefaces.

- JMS – Active MQ

Déposer chaque inscription dans un TOPIC JMS qui dispose d'un abonné.
Cet abonné sera matérialisé par une classe Java (Listener) situé dans une application Java standalone (ie : non-Web)

L'abonné écrira chaque message reçu dans un fichier de log propre à l'application. Utiliser l'implémentation ActiveMQ.

- OSGI

Utilisez la technologie OSGI pour permettre un remplacement à chaud du module core (jar) de l'application .

- JTA

Utilisez JTA au travers d'un plugin Tomcat de votre choix pour piloter la gestion transactionnelle.

- Alfresco

Mettre à disposition de l'utilisateur pour chaque formation un bouton lui permettant de télécharger le plan de la formation qui se situera dans une GED Alfresco.

- Spring Web

Réaliser la couche de contrôle à l'aide de l'implémentation Spring Web.

NB: Le code de cette partie n'est pas à rendre, cependant pendant les TP, l'application avec sa technologie complementaire devra être présentée au professeur.
La date de cette présentation de 5 mn sera fixée au dernier TP du semestre et sera rapellée dans l'agenda Moodle.