



Département Génie Informatique

Projets LO54 P2013

1) SUJET PRINCIPAL (MAIN REQUIREMENTS) :

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.

The application manages the training courses offered by a private school.

The school offers to people a Web application listing all the available courses and for each, the next training sessions planned.

The end user should be able to :

- filter the list of courses by a keyword potentially present in the title of the course*
- filter the list of courses by a date corresponding to the period of a training session*
- filter the list of courses by a location, the location coming from a dropdown menu.*

The end user will select a training session and subscribe to it providing with their personal information (First and Last name, Address, Phone, Email).

To simplify the model, we will consider that an end user subscribing to many training session will be registered many times in the DB.

It's not required to create an administration interface to manage the DB data.

On disposera donc des tables suivantes :

The database will be composed of the following tables :

LOCATION

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

COURSE

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

SESSION

- ID → Number AUTO (PK)
- START → Date Not Null
- END → 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
- SESSION_ID → Number Not Null (FK)

Use MySQL or Oracle.

2) TECHNOLOGIE COMPLEMENTAIRE (ADDITIONAL TECHNOLOGY) :

Par groupe de 3 max / 3 students per group (max)

Easy *

- BIRT

Ajouter 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

Add a function providing the user with a PDF document generated by BIRT regarding a customized template.

This PDF will display the list of the registered people per session

- JAX-RS

Ajouter à l'application un Webservice REST fournissant la liste des inscrits par session dans un format XML de votre choix.

Fournissez une application Java (non-Web) capable grâce à une implémentation JAX-RS de récupérer cette liste et de l'ajouter dans un fichier de log ou dans un écran simple.

Add to your application a REST Webservice providing with the list of registered people in a customized XML format.

Implement a new Java Application (non-Web) able thanks to a JAX-RS implementation to collect this list and to add it in a log file or to display it in a simple User interface.

- JPA 2

Exploiter JPA2 sur **son implémentation de référence(RI)** pour le requêtage.

*Use JPA2 over **its Reference Implementation (RI)** to proceed with the querying*

- JSF 2

Réaliser l'ensemble à l'aide du framework JSF 2 et sans librairies complémentaires (RichFaces, IceFaces et autres)

Use JSF 2 to implement this application. Don't add additionnal widgets (RichFaces, IceFaces...)

Medium **

- JMS – Active MQ

Déposer chaque inscription dans un TOPIC JMS qui dispose de 2 abonnés.
Le 1er abonné sera matérialisé par une classe Java (Listener) dans l'application Web.
Le second sera matérialisé par une classe Java (Listener) situé dans une autre application Java (non-Web)
Les 2 abonnés écrirons chaque message reçu dans un fichier de log propre à l'application.
Utiliser l'implémentation ActiveMQ.

*Add each new registration un a JMS TOPIC that is observed by to Listeners.
The first one is a class in the same Web application.
The second one is a class in a non-Web Java Application.
Both should write in a log file each message collected from the TOPIC.
Use the ActiveMQ implementation*

- JEE 6 / CDI

Utilisez une implémentation JEE 6 / CDI pour la configuration de l'architecture (IOC / Injection de dépendances)

Use a JEE6 / CDI implementation to configure your application architecture (IOC / Dependency / Injection)

- Spring Batch

Créer une tâche de fond Spring Batch que l'utilisateur puisse déclencher à partir d'une page spécifique.
Cette tâche de fond devra exporter vers un format XML de votre propre cru la liste des inscrits par session.

*Implement a Spring batch task a user can launch from a specific page.
This task should export in a customized XML format the list of the registered people per session.*

- Grails

Réaliser l'ensemble à l'aide du framework Grails

Use Grails to implement this application.

Hard ***

- Base NoSQL / Neo4J

Faites en sorte que le processus d'inscription s'effectue en double (dans vos méthodes Java) dans la base cible et dans une base NoSQL Neo4j. Un seul node doit être géré par Neo4j.

Improve your java code in order to double insert the registered people : in your initial database and in a NoSQL Neo4J database. One single node must be referenced by Neo4j.

- Spring ROO

Générez l'application cible grâce à Spring ROO.

Generate your application by the way of Spring ROO.

- 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.

Offer to the user the possibility to download the program of a course. Programs will be managed by an Alfresco instance.

3) LIVRABLES (DELIVERIES)

Dossier d'architecture logicielle et étude technique de la technologie complémentaire sélectionnée (Software architecture document and technical report about the additional technology) – Max 10 pages / PDF

- Mise en relation des composants logiciels et des flux d'information (*overview of the components and involved data flows*)
- Description de leur rôle vis-à-vis de l'architecture (*description of the role of these components regarding the architecture*)
- Technologies mise en oeuvre (*implemented technologies*)
- Fonctionnalités principales de la technologie complémentaire sélectionnée (*principle and main functions of the additionnal technology*)
- Difficultés rencontrées (*encountered issues*)

Archive et procédure d'installation (archive and deployment procedure)

- war ou autres archives (*.war or other archives*)
- Ressources externes : context.xml, fichiers de configuration et autres ressources externes statiques (*external ressources such as context.xml, configuration files and other static resources...*)
- procédure d'installation – readme.txt (*deployment procedure – readme.txt*)

Code source (*source code*)

- Applications Java (*Java applications*)
- Applications Web Dynamiques (*Dynamic web applications*)
- *Autres projets eclipse* (Other Eclipse projects)

Important / Important notes :

- Ne regardez pas les écrans outre mesure. *Don't provide with advanced look and feel / design within the UI.*
- Ne passez pas plus de 30 heures sur le projet. *Don't spend more than 30 hours for this project.*
- Une soutenance orale de 20 minutes sera prévue courant Juillet durant laquelle vous présenterez en quelques slides votre projet. *A 20 minutes presentation will be planned on July while you will get the opportunity to present in few slides your project.*