

**Projet Java EE**  
**Portail étudiant**  
**A rendre pour le 2011 à 18h.**

*L'ensemble du code, des noms de classes, des champs, des méthodes doivent être en anglais. Le code doit répondre aux normes SUN et être obligatoirement testé unitairement.*

*Le rendu du projet sera fait par mail à l'adresse suivante: [loyaute@univ-mlv.fr](mailto:loyaute@univ-mlv.fr).*

Le but du projet est de réaliser un portail étudiant. Ce portail va permettre à différents utilisateurs de réaliser des actions administratives ou académiques. Vous devez mettre particulièrement l'accent sur le découpage en couche de l'application (données, service et présentation), la propre du code et l'utilisation des frameworks et API à votre disposition.

**1. Les acteurs:**

Il va y avoir 4 types d'acteurs pouvant interagir dans le portail. Ces types sont les suivants:

- administrateur du portail;
- étudiant;
- professeur;
- administratif.

Les acteurs vont avoir différentes actions possibles dans ce portail.

**2. Les actions de l'administrateur:**

C'est l'administrateur du portail. Ses droits sont globales.

**3. Les actions des étudiants:**

Un étudiant va pouvoir réaliser les opérations suivantes:

- consulter (en ligne ou au format PDF) un sujet de TD, TP ou projet;
- consulter au format PDF les cours;
- rendre (uploader dans le portail) un TP, TD ou un projet;
- s'inscrire à un tronc commun pour son année scolaire;
- s'inscrire dans des options pour son année scolaire;
- consulter les notes dans chacune des matières à laquelle il s'est inscrit;
- consulter les dates d'examen;
- consulter les dates de consultation de copie;
- consulter (en ligne ou au format PDF) les annales des années précédentes;
- consulter son avis (ou non) de passage dans l'année supérieure;
- saisie non nominative des avis sur chaque matière (questionnaire + commentaire libre);
- consultation du planning rendu TP, TD, projet et examen;
- consultation de son planning;
- consultation du planning des salles pour ses matières;

**4. Les actions des professeurs:**

Un professeur va pouvoir réaliser les opérations suivantes:

- saisir au sein de l'application ses sujets de TP, TD, projet et examen (attention on peut saisir en plusieurs fois le sujet puis le "poster" sur le site, i.e. le rendre consultable);

- mettre à disposition ses slides de cours;
- consulter (en ligne ou au format PDF) ses TD, TP, projet;
- consulter (au format PDF) ses cours;
- fournir un descriptif de son cours;
- fournir une liste de mots clé de son cours;
- consulter la liste de ses étudiants;
- marquer un TD, TP ou projet comme nécessitant un rendu et affecter une date de rendu;
- affecter une note pour chaque TD, TP, projet nécessitant un rendu (attention au non rendu...) et examen;
- calculer automatiquement la moyenne pour un TD, TP, projet ou examen;
- calculer automatiquement la moyenne pour chaque étudiant;
- produire une version PDF de la liste des notes pour chaque TD, TP, projet et examen;
- produire une version PDF de la liste des moyennes pour sa matière (distinguer contrôle continu, projet, examen s'il y a lieu);
- consultation du planning rendu TP, TD, projet et examen;
- consultation du planning des salles;
- consultation de son planning;
- réservation d'un vidéo projecteur;

Un professeur peut enseigner plusieurs disciplines.

#### 5. Les actions des administratifs:

Un administratif va pouvoir réaliser les opérations suivantes:

- constitution des portefeuilles de matières d'un tronc commun;
- constitution des portefeuilles de matières optionnelles;
- constitution des règles d'incompatibilités entre options et options-tronc commun;
- saisies des dates d'examen pour toutes les matières;
- saisie des informations des étudiants;
- saisie des informations des professeurs;
- saisie des avis de passage des étudiants;
- consultation du planning rendu TP, TD, projet et examen;
- affectation des salles (cours, TD, TP) aux différentes matières (signaler les conflits possibles);
- consultation du planning des salles (signaler les conflits possibles);
- gestion de l'affectation des vidéo projecteurs;

#### 6. Enrichissement des fonctionnalités

Vous pouvez enrichir cette application avec d'autres fonctionnalités qui vous semblent pertinentes.

#### 7. Technologies utilisées:

- Java 1.6;
- Authentification via LDAP ou DB;

- Gestion de la sécurité via Spring-security;
- Birt pour l'édition PDF des documents;
- log4j pour logger l'application;
- MySQL ou PostgreSQL pour la DB;
- Maven2;
- SVN/CVS comme gestionnaire de code source;
- CSS pour l'apparence des pages;

#### 8. Modalité de rendu:

Le rendu sera fait sous forme d'un fichier zip contenant:

- étude préalable;
- scripts SQL de création et d'initialisation de la DB;
- documentation utilisateur;
- planning de consommé (ce qui a été fait, quand, pendant combien de temps, etc. **non nominatif!**);
- war de l'application;
- le code source et les tests (unitaires, fonctionnels, de charge, etc.);
- documentation d'installation (démarrage de l'application, etc.);
- documentation technique (architectures, diagramme de classes, diagramme des modules maven, API utilisés, choix techniques, etc.);
- planning prévisionnel (rendu durant le 29 novembre 2010);
- chiffrage du projet (voir ci-après);

#### 9. Chiffrage (hors taxe) du projet:

- TJM test fonctionnel: 600 euros
- TJM développeur: 500 euros
- TJM architecte: 750 euros
- TJM spécification: 600 euros
- TJM chef de projet: 800 euros