

## Projet de Bases de données-2

### Application de gestion de planning (Karim LAHLOU)

---

#### **Instructions générales**

Ce programme de gestion de planning de cours devra être réalisé entièrement en Java ou tout autre langage en accord avec votre chargé de TP (C#, PHP, Python, etc...) et la base de données Oracle, MySQL, PostgreSQL, Derby, HsqlDb, SqlServer ou Access.

#### **Vous pouvez vous aider :**

- ✓ Du support de cours (incontournable)
- ✓ Des ressources disponibles sur Internet, notamment sur le site d'Oracle
- ✓ De l'encadrement en TP/Projet.

#### **Modules, API et concepts concernés par ce programme (à titre d'information)**

- Concepts généraux de Java (héritage, polymorphisme, encapsulation, surcharge, méthodes...)
- Constructeurs, interfaces, classes abstraites,
- *Garbage collection*
- Collections
- Threads
- JDBC et Gestion de la persistance
- Interfaces graphiques et Swing
- ORM Hibernate ou JPA, ...

#### **Critères d'évaluation**

- Ergonomie (vous avez carte blanche sur le choix du '*look and feel*' de l'IHM). Mais votre interface doit être visuellement agréable, "*user friendly*" et intuitive en fenêtre graphique ou en console en lignes de commande.
- Qualité du code (commentaires, algorithmes utilisés)
- Respect du pattern MVC souhaité
- Gestion des exceptions
- Gestion optimale, flexible et simple de la persistance
- Richesse des concepts mis en œuvre et des API utilisées.
- Respect du cahier des charges

#### **Etapes de mise en œuvre**

- 1) Avant tout, imprégnez-vous des exigences du cahier des charges
- 2) Générez des maquettes de l'IHM. Pour ce, vous pouvez par exemple utiliser au choix l'un des sites suivants :  
<http://lumzy.com/app/>  
<https://gomockingbird.com/mockingbird/>
- 3) Rédigez un mini document de conception technique avec le dictionnaire de données, le MCD, MLD et/ou MPD, un diagramme de classes souhaité pour les classes Java voir aussi un diagramme de cas d'utilisation et/ou séquences élémentaire. Le but est d'illustrer le choix de vos entités amenées à être implémentées sous forme de classes et les interactions entre ces entités.



## **Cahier des charges**

### ***I – Introduction***

Ce programme est destiné au service planning qui souhaite gérer la planification des cours dans l'établissement.

Il permettra deux types d'actions :

- ✓ La gestion administrative simple des élèves, des cours, des professeurs et des salles.
- ✓ La visualisation du planning par chaque élève et la possibilité de le mettre à jour par le personnel du service planning.

### ***II – Fonctionnalités techniques***

- ✓ N'oubliez pas de créer votre package spécifique (ne pas utiliser -par exemple- le package par défaut de NetBeans ou Eclipse).
- ✓ Lors de la fermeture du programme, un message de confirmation doit être proposé à l'utilisateur.
- ✓ Un couplage faible avec le SGBD est impératif. Ceci permettra entre autres d'inter changer de SGBD avec un minimum de codage et de paramétrage.
- ✓ Pour la gestion de la persistance, l'utilisation des frameworks Hibernate ou JPA est autorisée. Toutefois nous attirons votre attention sur le point concernant le couplage faible.
- ✓ Libre à vous de choisir l'un des SGBD suivants pour implémenter votre base. Il faudra juste préciser la version que vous utilisez dans votre document technique. Pour tout autre SGBD, bien vouloir nous prévenir à l'avance.
  - a. Oracle
  - b. MySQL
  - c. PostgreSQL
  - d. Apache Derby (alias Cloudscape ou Java DB).
  - e. HSQLDB
  - f. SqlServer ou Access
- ✓ Toutes les ressources ouvertes doivent être libérées au pire à la fermeture du programme.

### ***III – Gestion administrative des élèves, des cours, des groupes, des professeurs et des salles***

Tous les champs ci-dessous doivent être modifiables, sauf indication contraire.

- **Général :**
  - Matricule (non modifiable et doit être auto incrémenté)
  - Nom
  - Prénom
- **Identité :**
  - Date de naissance
  - Ville de naissance
  - Pays de naissance
  - Sexe
  - Date d'inscription
  - Etablissement précédent
  - Photo (possibilité de télécharger une photo présente sur le poste de travail)
- **Coordonnées**
  - N° et Nom de rue
  - Code postal
  - Ville

- Tél domicile
- Tél mobile
- Email
- **Personnes(s) responsable(s)**  
Prévoir au moins **un** contact.
  - Nom
  - Prénom
  - Adresse
  - Téléphone
  - Email
- **Les professeurs :**
  - Matricule (non modifiable et doit être auto incrémenté)
  - Nom
  - Prénom
  - N° et Nom de rue
  - Code postal
  - Ville
  - Tél domicile
  - Tél mobile
  - Email
- **Les salles :**
  - Code (non modifiable et doit être auto incrémenté)
  - Nom
  - NbPlaces
  - TypeSalle : Amphi, Cours, Labo, SalleSoutenance

#### ***IV – Gestion du planning des élèves***

Attention à bien gérer les différentes années scolaires !

- Recherche de listes par promotion ou groupe de TD (Les élèves sont organisés en groupes de Td/Tp/Projet et font partie d'une promotion L1, L2, L3, M1, M2). Nous considérons que tous les élèves d'un groupe suivent toujours le même cours sur un horaire donné.
- Les cours sont prévus du lundi au samedi de 8h00 à 20h00 en créneaux de 2 heures avec les contraintes suivantes :
  - Un élève ne doit pas avoir plus de 8 heures par jour et 30 heures par semaine.
  - Un élève doit avoir au minimum une pause de 12h00 à 14h00 ou de 14h00 à 16h00 pour pouvoir déjeuner.
  - Une salle ne peut pas recevoir plus d'élèves que sa capacité maximale.
  - Pour une meilleure gestion des pauses, les cours sont réellement programmés sur des créneaux de 1h45mn et ceci permet de disposer de 15 minutes entre 2 horaires successifs pour mieux gérer les flux et permettre une pause de 15 minutes aussi.
- Gestion de planning sur une semaine ou entre deux dates pour un élève.

#### ***V – Fonctionnalités avancées***

Ce défi rapportera jusqu'à **4 pts sur la note finale** aux membres du groupe

Seule condition : **Toutes** les fonctionnalités de base doivent être mises en œuvre.

- ✓ Login avec trois niveaux de privilèges : **Admin, Prof** et **Eleve**.
  - L'administrateur aura des droits complets de mise à jour et la planification des cours.
  - Le prof a le droit de consulter son planning et de modifier sa salle par une autre salle qui doit être disponible au créneau souhaité.
  - L'élève peut consulter son planning.
  - Tout le monde peut consulter les salles disponibles pour une journée donnée.

- ✓ Possibilité à deux clients "administrateurs" de se connecter simultanément à la base ce qui implique de votre part la mise en œuvre d'une gestion transparente et optimale de l'accès concurrentiel au SGBD.

## **VI – Fonctionnalité bonus**

Chacun de ces deux défis rapportera jusqu'à **2 pts sur la note finale** aux membres du groupe.

- ✓ Génération de l'emploi du temps en format pdf ou csv.
- ✓ Gestion des absences d'un élève sur un mois donné.

## **Consignes d'organisation**

- S'organiser en groupes de 2 membre minimum et 3 membres maximum par groupe (liste à remettre à la fin de la séance 4 du cours/TD/TP/Projet)
- Utiliser un AGL (Atelier de Génie Logiciel) pour élaborer votre schéma conceptuel de données (MCD) et générer le schéma logique puis le schéma physique de votre base de données ainsi que le code SQL pour la création de votre base de données (JMerise par exemple).
- Intégrer à votre script toutes les contraintes d'intégrité nécessaires pour le bon fonctionnement de votre application.
- Proposer un jeu d'essai en remplissant quelques tables pour valider votre bonne conception.
- Créer et testez votre base de données.
- Développez votre application en utilisant un EDI (Environnement de Développement Intégré) en testant d'abord la communication avec votre base de données puis en implémentant les cas d'utilisation que vous avez choisis. Vous pouvez utiliser aussi un AGL comme StarUML pour vous faciliter la conception de votre application.
- Pensez à définir des priorités pour garantir la livraison de votre projet avec un minimum de fonctionnalités dans les délais.
- Séance de validation du travail fini à la dernière séance de Tp/Projet.
- Un rapport décrivant votre projet en intégrant le dictionnaire de données, MCD, MLD ou MPD et voir aussi le diagramme de classes et/ou cas d'utilisation ainsi qu'un scénario d'exécution avec des prises d'écran et en mettant en évidence les fonctionnalités implémentées, celles envisagées et non terminées et aussi les difficultés rencontrées et les points forts et faibles du projet.
- Le rendu exclusivement sur campus doit se faire avant le dimanche 23/12/2018 à 23h55.

## **Rendus attendus**

- 1) Au plus tard, le **23/12/2018 à 23h55**, vous devez déposer vos livrables dans un fichier zip, sur Campus (Projet source Java + scripts SQL + Rapport).
- 2) Tous les fichiers sources java, scripts SQL et documents doivent contenir les noms des membres du groupe en en-tête.
- 3) Les livrables seront importés sur votre projet dans Eclipse. Objectif "zéro erreur" à ce niveau.
- 4) La documentation comptera pour 20% de la note du projet et sera évaluée minutieusement avant même le lancement de votre programme.
- 5) Au départ, chaque groupe disposera d'un capital maximum de 20 pts. Au fur et à mesure que des erreurs, des "oublis" ou des imperfections seront rencontrées, un ou des points seront décomptés.