

## **TP-3 : Etude de cas Gestion Formation**

### **Diagramme de Classes en MVC – Diagramme d'états-transitions**

### **Diagramme de composants – Diagramme de déploiement**

## **I - Introduction**

Le but de ce TP est de réaliser une application Orientée Objet pour la gestion des notes des étudiants d'un centre de formation en C++.

Pour cette application, on vous fournit un cahier des charges et à vous de faire la phase d'analyse en utilisant la notation UML et la conception orientée objet dans le but de l'implémenter en C++.

## **II – Cahier des charges de l'application GestNotes :**

### **Contraintes Techniques**

1. L'application gère de préférence des tables de hachage utilisant le hachage externe.
2. A la fermeture de l'application, les données en mémoire sont enregistrées sur un espace persistant (fichiers, BDD, ...)

### **Gestion des notes**

Toute note est matérialisée sous la forme d'un quadruplet (TP, PRJ, DE).

Le chef du centre de formation souhaite l'informatisation de la gestion des notes de ses élèves. Il veut avoir une application qui permet d'inscrire de nouveaux élèves et de les affecter à un groupe de spécialité. Toute spécialité se déroule sur 3 années et elle a une liste de modules obligatoires. A chaque module, est affecté un professeur responsable du module ainsi que plusieurs assistants qui sont des professeurs ou des élèves des classes supérieures. Tout élève est rattaché à un groupe de spécialité et suit tous les modules obligatoires de sa spécialité. Les élèves suivent aussi des modules optionnels. Les modules optionnels sont choisis par l'élève parmi les modules de même année des autres spécialités. Le passage à l'année suivante est conditionné par l'obtention de la moyenne de 10/20 sur chacun des cours obligatoires et une moyenne générale de 10/20 sur l'ensemble des cours obligatoires et optionnels inclus. Chaque élève a 5 à 7 modules obligatoires par année d'étude et doit s'inscrire à exactement 3 modules optionnels parmi l'ensemble des modules de même année d'étude de toutes les autres spécialités. A chaque module, sont attribuées des notes (DE, TP, Projet), ainsi qu'une pondération par note, et une note finale sur 20 représentant la moyenne du module. Si un TP ou Projet n'est pas intégré dans la notation d'un module, il aura sa pondération à 0. A

chaque élève est affecté un professeur tuteur qui le suit dans sa scolarité. Ce dernier est alerté automatiquement dès que la moyenne d'un module est en dessous de 10.

Les élèves et les professeurs sont inscrits par un membre de l'administration du centre de formation ainsi que l'affectation d'un professeur à un module ou d'un professeur tuteur à un élève et l'affectation d'un élève à une spécialité ou à l'admission en année supérieure.

Les notes sont saisies ou mise à jour uniquement par le professeur responsable du module et la moyenne est générée automatiquement par l'application.

## Gestion des élèves

GestNotes doit proposer tous les types de recherche et d'affichage suivants pour les élèves :

- Liste par ordre alphabétique sur l'ensemble des élèves;
- Liste des élèves par groupe;
- Recherche par nom;

L'application doit pouvoir permettre d'insérer ou supprimer un élève ou modifier ses informations personnelles (adresse, téléphone, etc.).

## Gestion de la persistance des données

Les fichiers de données ou bases de données doivent contenir des données "réelles" : Elèves, Professeurs, Modules, etc...

## Interface Graphique

### Affichage /saisie

- GestNotes doit proposer des affichages en couleur.
- Toutes les saisies effectuées doivent être sécurisées.
- Vous êtes seuls maîtres des informations affichées/à saisir : pensez à l'ergonomie de votre application !

### Raccourcis clavier

- Des touches clavier peuvent être associées comme raccourcis aux actions importantes de votre application.

## III – Organisation :

Le travail est à faire **au maximum** en trinôme et **au minimum** en binôme.

## IV – Travail à effectuer :

Vous devrez modéliser en UML l'analyse de l'application GestNotes dans le but de son implémentation en C++ :

---

### Etape 1

Etablir un glossaire lié au système à étudier

---

### Etape 2

Etablir le modèle des cas d'utilisations (**Diagramme de cas d'utilisation**) du système.

- Identifier les cas d'utilisations (recenser les fonctionnalités de base du système et les liens entre cas d'utilisation)
-

- 
- Identifier les acteurs du système (distinguer les acteurs principaux des acteurs secondaires)
  - Etablir le diagramme des cas d'utilisation
- 

### **Etape 3**

---

Donner une description textuelle d'un cas d'utilisation au choix

### **Etape 4**

---

Détailler le scénario nominal du cas d'utilisation choisi avec un **diagramme de séquence (la vision boîte blanche est préconisée)**

### **Etape 5**

---

Etablir le **diagramme de classes de niveau métier** + Les classes identifiées dans l'étape précédente

### **Etape 6**

---

Etablir le **diagramme d'états-transitions** d'un élève.

### **Etape 7**

---

Etablir le diagramme de composants.

### **Etape 8**

---

Etablir le diagramme de déploiement.

### **Etape 9**

---

Générer le code C++ associé à partir du diagramme de classes.

## **V - Timing**

La remise du compte-rendu doit se faire par dépôt sur e-campus le soir de la séance avant 22h55.

Le compte-rendu en format pdf contenant les réponses aux différentes étapes demandées.

Le format TP3UML\_NOM1\_NOM2 où NOM1 et NOM2 sont respectivement les noms des participants.