

## NoSQL – TP2

L'objectif de ce TP est de manipuler une base de données MongoDB depuis une application Java.

On reprend la version **Swing** de l'application de **gestion de notes réalisée dans le TP7** de la ressource **R3.04** (Qualité de développement) et on choisit d'utiliser la base de données et les requêtes définies dans le TP précédent (NoSQL-TP1) pour la modélisation des données.

### **Partie dialogue**

Pour le dialogue avec l'utilisateur, on reprend les ihms du QD-TP7 (cf. annexe 1) : une fenêtre principale « Gestion notes » permettant de saisir l'identifiant et le mot de passe et de se connecter ; une fenêtre « Session enseignant » relative à une session enseignant, qui permet de sélectionner une matière enseignée par l'enseignant connecté et de modifier ou de supprimer les notes des étudiants inscrits dans cette matière ; une fenêtre « Session étudiant » relative à une session étudiant permettant d'afficher les notes de l'étudiant connecté. Le diagramme des classes des ihms est montré par l'annexe 2. Ces classes sont rassemblées dans un package nommé `ihms`.

### **Partie données de l'application**

Les classes décrivant les données définies dans QD-TP7 sont modifiées afin de pouvoir traduire les documents de la base de données en objets Java correspondants à l'aide de Gson (cf. annexe 3). Ces classes sont rassemblées dans un package nommé `donnees`.

### **Partie manipulation des données**

L'ensemble des données est mémorisé dans la base de données réalisée dans NoSQL-TP1. Pour accéder à cette base et manipuler son contenu, on introduit les classes suivantes rassemblées dans un package nommé `utilitaires_ihms` (cf. annexe 4) :

- Classe **ConnexionBD** : Elle permet d'obtenir un accès à la base de données.
- Classe **GestionVerrou** : Elle définit les opérations d'accès concurrent : verrouillage et déverrouillage de l'application, vérification de la présence du verrou.
- Classe **GestionNotes** : Elle définit les opérations nécessaires au fonctionnement des ihms :
  - Chargement des données à partir de la base et traduction en objets Java.
  - Vérification de l'identifiant et le mot de passe d'un utilisateur qui demande à se connecter.
  - Recherche des données et modification des notes.

### **Travail à réaliser**

N.B. : Pour les méthodes nécessitant l'exécution de requêtes MongoDB, on utilisera les requêtes définies dans NoSQL-TP1 (correction à télécharger depuis Moodle) en les adaptant si nécessaire.

**1.** Ouvrir **Netbeans 8.2** et créer un nouveau projet **Java Application**. Dans l'arborescence du projet faire un clic-droit sur **Librairies**, sélectionner **Add JAR/Folder...** puis ajouter les `.jar mongo-java-driver` et `gson` (à télécharger préalablement depuis Moodle).

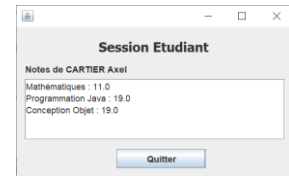
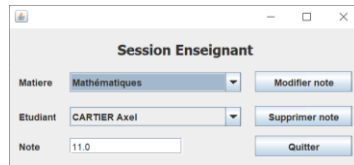
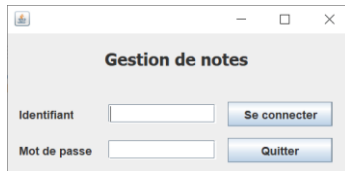
**2.** Créer trois packages nommés `donnees`, `ihms` et `utilitaires_ihms`. Télécharger le code à compléter (`nosql_tp2_code_a_completer.zip`) depuis Moodle et copier le code source dans les packages correspondants.

**4.** En se basant sur les diagrammes de l'annexe 4, compléter les méthodes « TO DO » des classes `ConnexionBD` et `GestionNotes`.

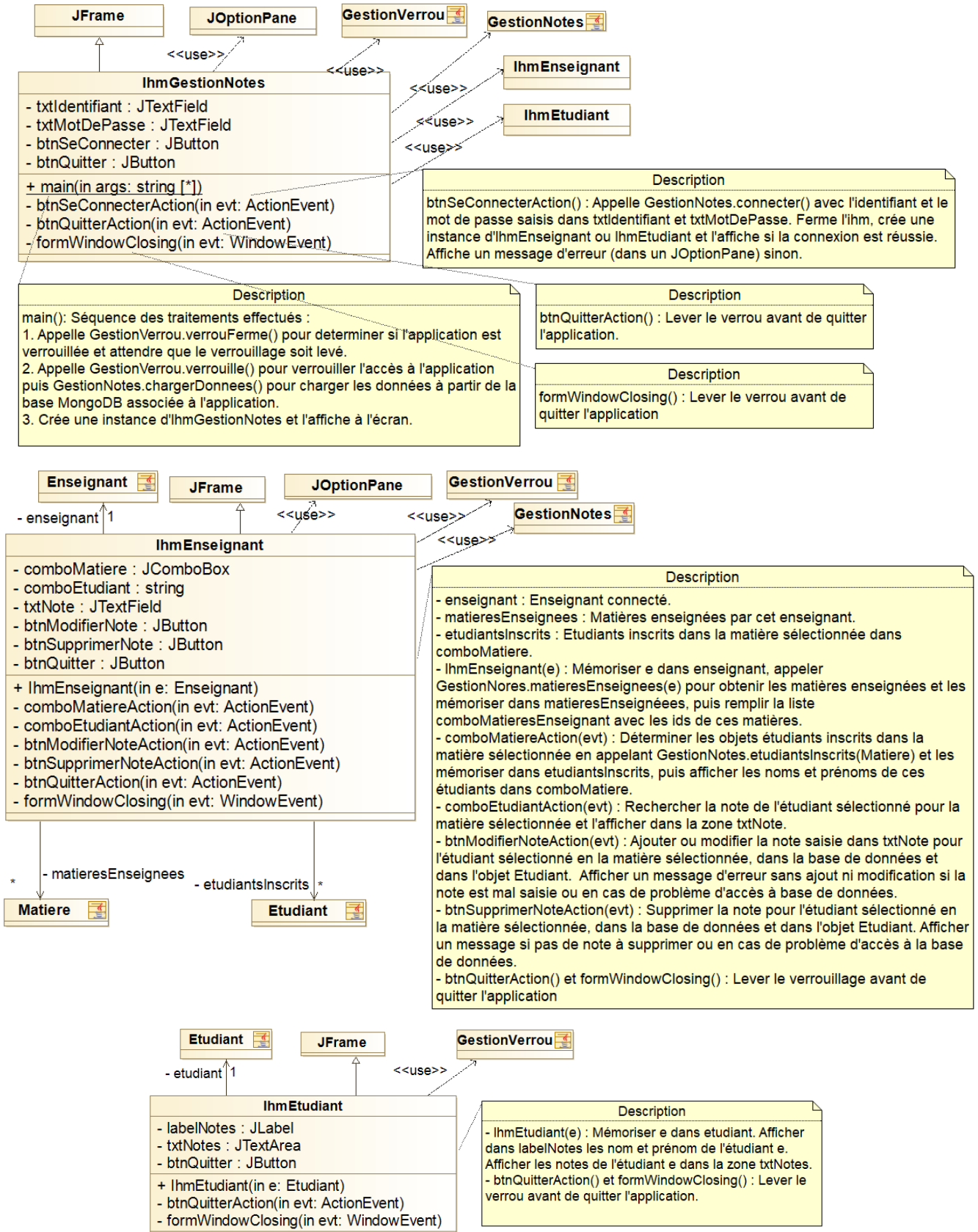
**5.** Ouvrir MongoDB Compass, cliquer sur Connect. Supprimer la base de données créée dans le TP précédent (NoSQL-TP1) puis la recréer et la peupler à l'aide des requêtes de la question 3 du TP (utiliser la correction à télécharger sur Moodle si nécessaire).

**6.** Tester l'application et utiliser MongoDB Compass pour vérifier les modifications des données.

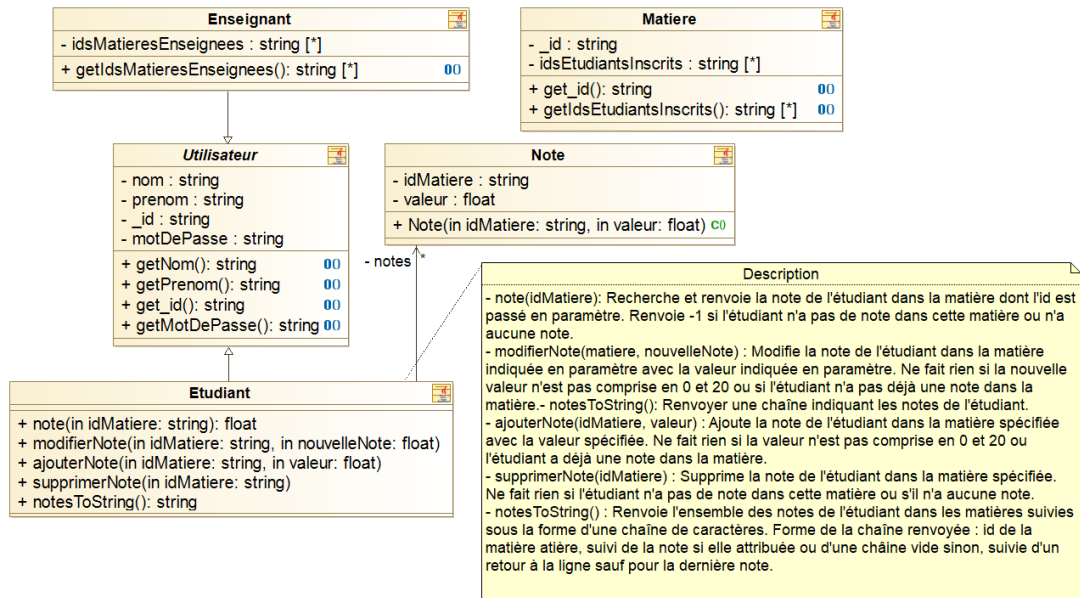
## ANNEXE 1 : Fenêtres de dialogue



## ANNEXE 2 : Diagramme de classes des ihms (package ihms)



### ANNEXE 3 : Diagramme de classes des données (package donnees)



### ANNEXE 4 : Diagramme de classes de manipulation des données (package utilitaires\_ihms)

