

TP Technologies Web

Ilyas Taoussi (ICY)

Mr. Degroote

Sommaire

1 – Schéma de la base de Données

2- Schéma et Fonctionnement de L'application

3- Conclusion

4- Source

la Base de données lié à l'application

Consignes du TP à Respecter :

- Utiliser HSQLDB (respectée)
- Plusieurs Entités
- Au moins une relation entre 2 entités

Entités de La base de Données :

-Etudiant , Professeur , Logins , Specialite , Module , Note

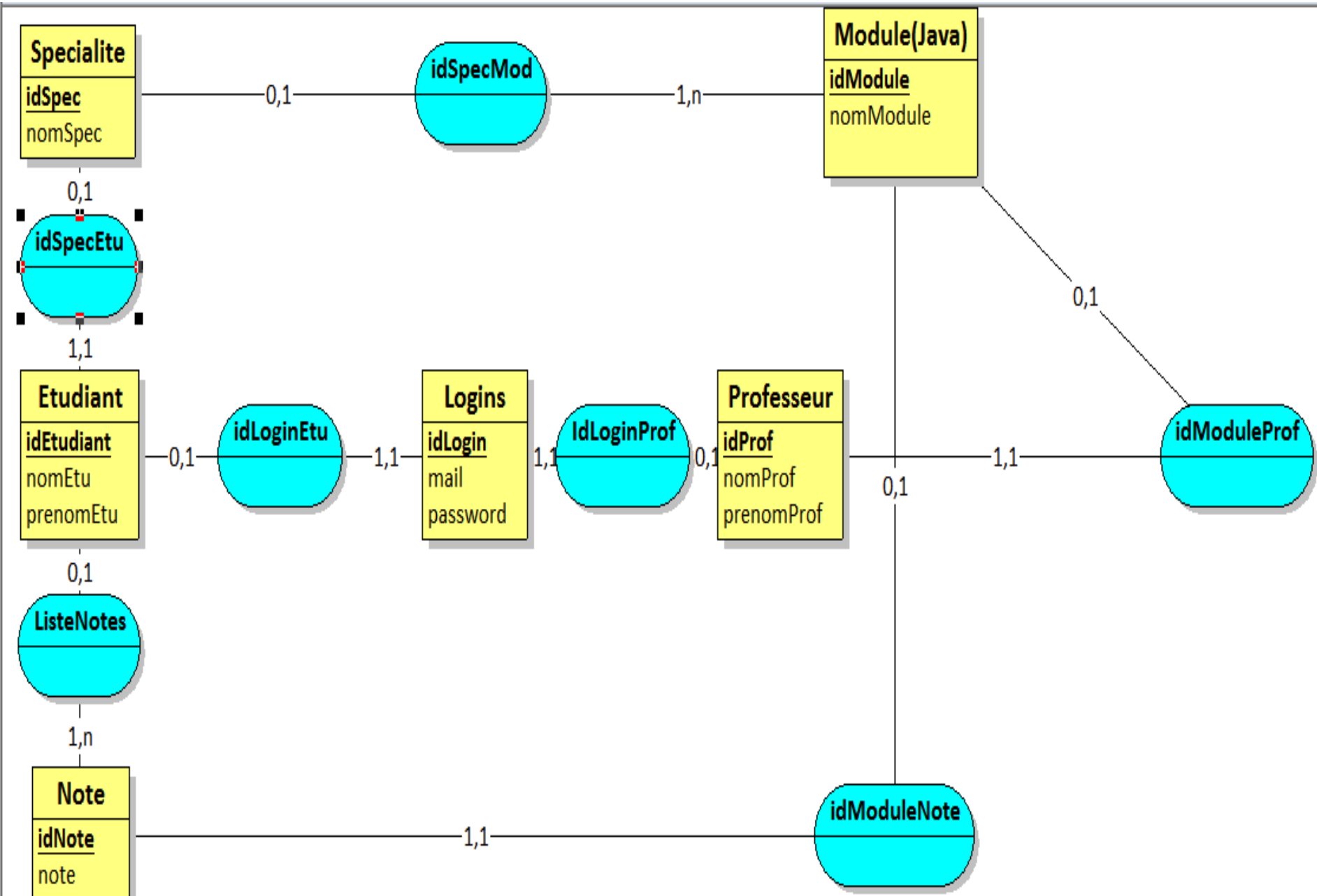
⇒ **Consigne Respectée !**

Relations entre les Entités :

- @OneToMany : Etudiant -> List<Note>
- @OneToOne : Etudiant -> Logins , Professeur -> Logins / Module, Note -> Module
- @ManyToOne: Etudiant -> Specialite , Module -> Specialite

⇒ **Consigne Respectée !**

Schema de la BDD (modelisation avec Looping)



Fonctionnement de l'application :

- 1- Presentation des classes implementées dans l'application**
- 2- Administrateur**
- 3 -Services Communs/Particuliers : Professeur , Etudiant**

Présentation des classes et Fichiers de l'Application

(Propres Classes et Fichiers)

Java (18 classes et 6 Interfaces):

- 1- Entities : 6 (mentioned before)
- 2-CRUD Repositories : <Entity>Repository (6 in Total)
- 3- REST services : <Entity>REST (6 in Total)
- 4- Else :
 - a- Input : EtudiantInput , LoginInput , ModuleInput , ProfInput
 - b- Utilities : **HashClass , MainController**

HTML : 12 document

Javascript : 8 fichiers

CSS : 1 fichier (own.css)

Classes Particulières :

-HashClass : Classe contenant 2 méthodes statiques pour avoir le hash du mot de passe récupéré et l'enregistrer dans la base de données

```
public static String StringToSHA256Hash(String StringToHash) throws Exception {  
    MessageDigest md = MessageDigest.getInstance("SHA256");  
    byte[] digest = md.digest(StringToHash.getBytes(StandardCharsets.UTF_8));  
    return bytesToHex(digest);  
}
```

-LoginsRepository: le CRUD Repository de la classe Logins , contenant 2 méthodes définies avec une @Query personnalisée

```
public interface LoginsRepository extends CrudRepository<Logins, Integer>{  
    @Query("select l from Logins l where l.mail = :mail and l.password = :password ")  
    Optional<Logins> findByMailAndPassword(@Param("mail") String mail,@Param("password") String password);  
}
```

Services Administrateur

Demonstration sur le site :

- Ajout d'une specialité , Ex : Light Novel
- Ajout d'un Module , Ex : No Game no Life -> Light Novel
- Ajout d'un Etudiant , Ex : Ouad Sami -> Light Novel
- Ajout d'un Professeur , Ex: Light Novel - > NGNL -> (author)

Services Utilisateur : Professeur / Etudiant

Démonstration :

1 - Services Communs : Connexion , Inscription, Déconnexion

2 - Services Particuliers :

A – Professeur (Ajouter des Notes , Profil)

B -- Etudiant (Consulter les Notes , Profil)

Conclusion

- Exigences du TP bien respectées (Spring Boot , REST Jersey , JS/jQuery, HSQLDB)
- Temps Insuffisant pour tester d'autres capacités (surtout pour un travail seul)
- Idées Pour d'autres services pour l'application
- Fichiers Importés : CSS (html5up) , Page Admin (Nicolas/Quentin)

Sources:

Fichiers CSS (except own.css): <https://html5up.net/>

Depot GitHub : <https://github.com/IlyasTaoussi/TP>