**PROJET ECOLE**

**Objectif General :**

Créer une application de gestion d’une école.

**Description du besoin:**

le système doit prendre en compte des options suivantes :

1. Gestion étudiants :

* CRUD pour étudiants
* Rechercher multicritère par notes et semestre

1. Gestion des écolages :

* L’écolage peut être paye plusieurs fois en une semestre
* L’écolage est différente pour chaque niveau

1. Gestion des notes :

* Enregistrement des notes
* Génération des bulletin à chaque fin de semestre ou fin de l’année
* Validation des notes des étudiants par le professeurs
* Résultat à la fin de l’année par classe
* Générer un PDF de notes d’un étudiants

**Règles de Gestion :**

* + - Une année scolaire comporte un ou plusieurs examens
    - Un examen comporte plusieurs matières
    - Un étudiant ne peut pas payer un écolage supérieur s’il n’a pas encore paye l’écolage récent
    - Lors de l’insertion d’un écolage le système vérifie s’il ne numéro étudiant existe
    - Une matière a un coefficient
    - Mérite:

|  |  |
| --- | --- |
| Moyenne des notes |  |
| 10 <= N < 12 | Passable |
| 12<=N<14 | Assez bien |
| 14<= N < 16 | Bien |
| 16<=N < 18 | Très bien |
| N<=18 | Excellent |

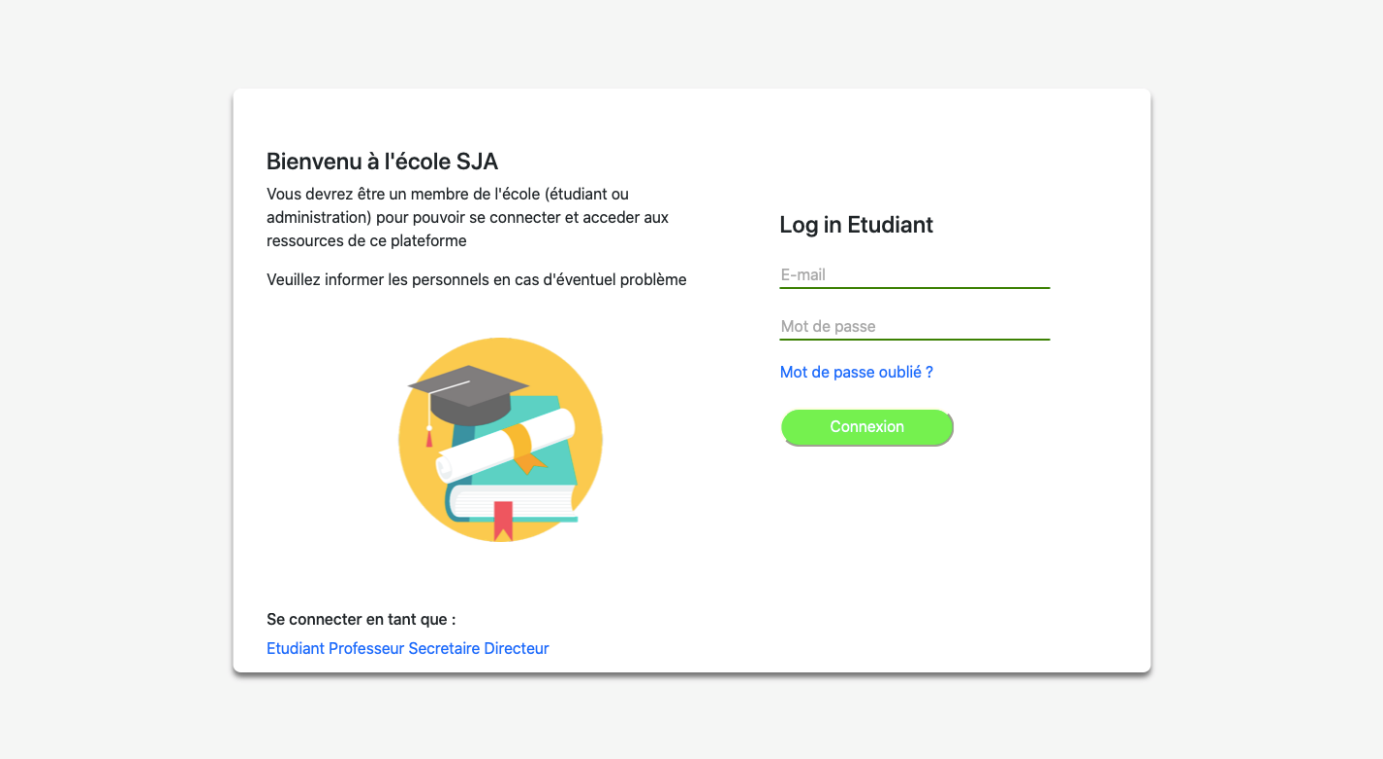
**Développement :**

**A ) AFFICHAGES :**

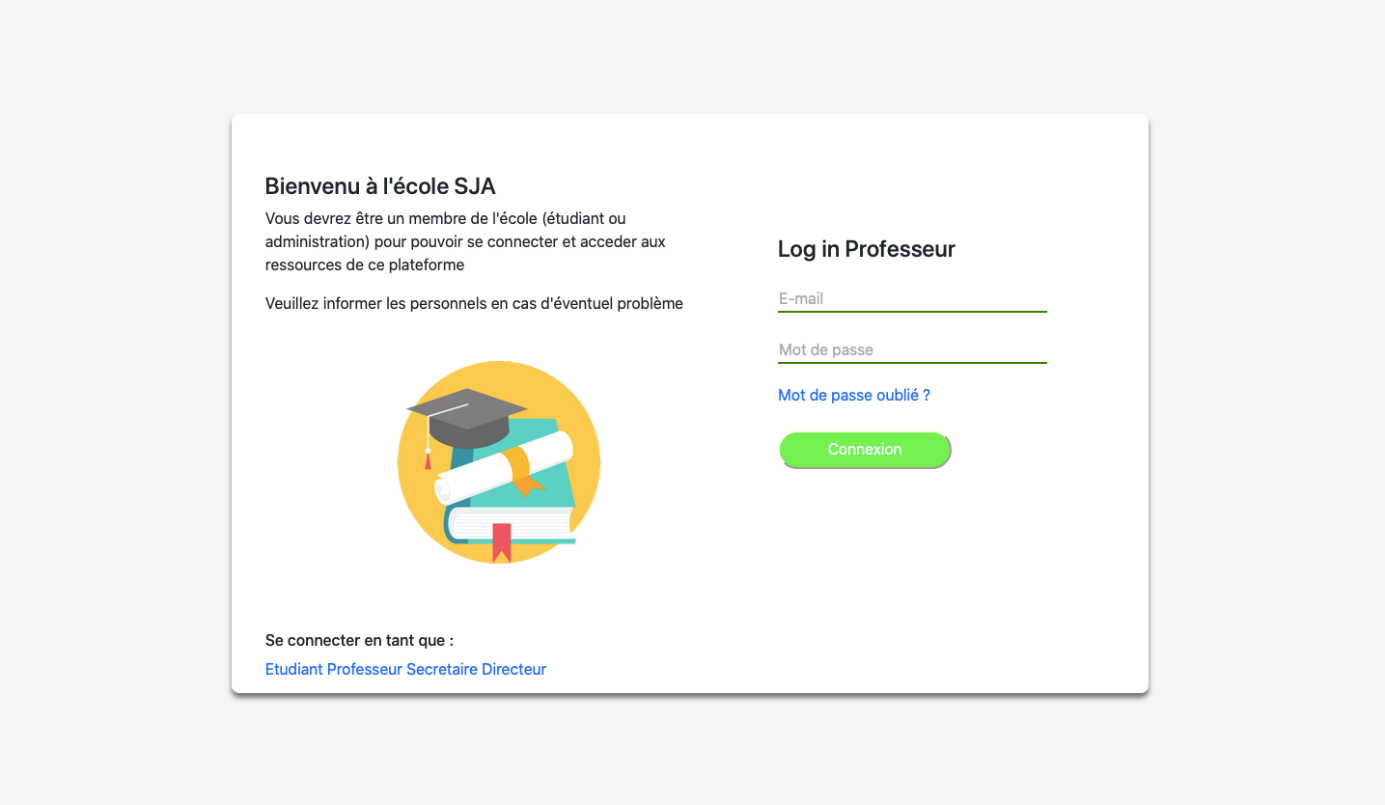
Voici les affichages à faire pour le logiciels pour gérer l’école :

1. Login Étudiants :

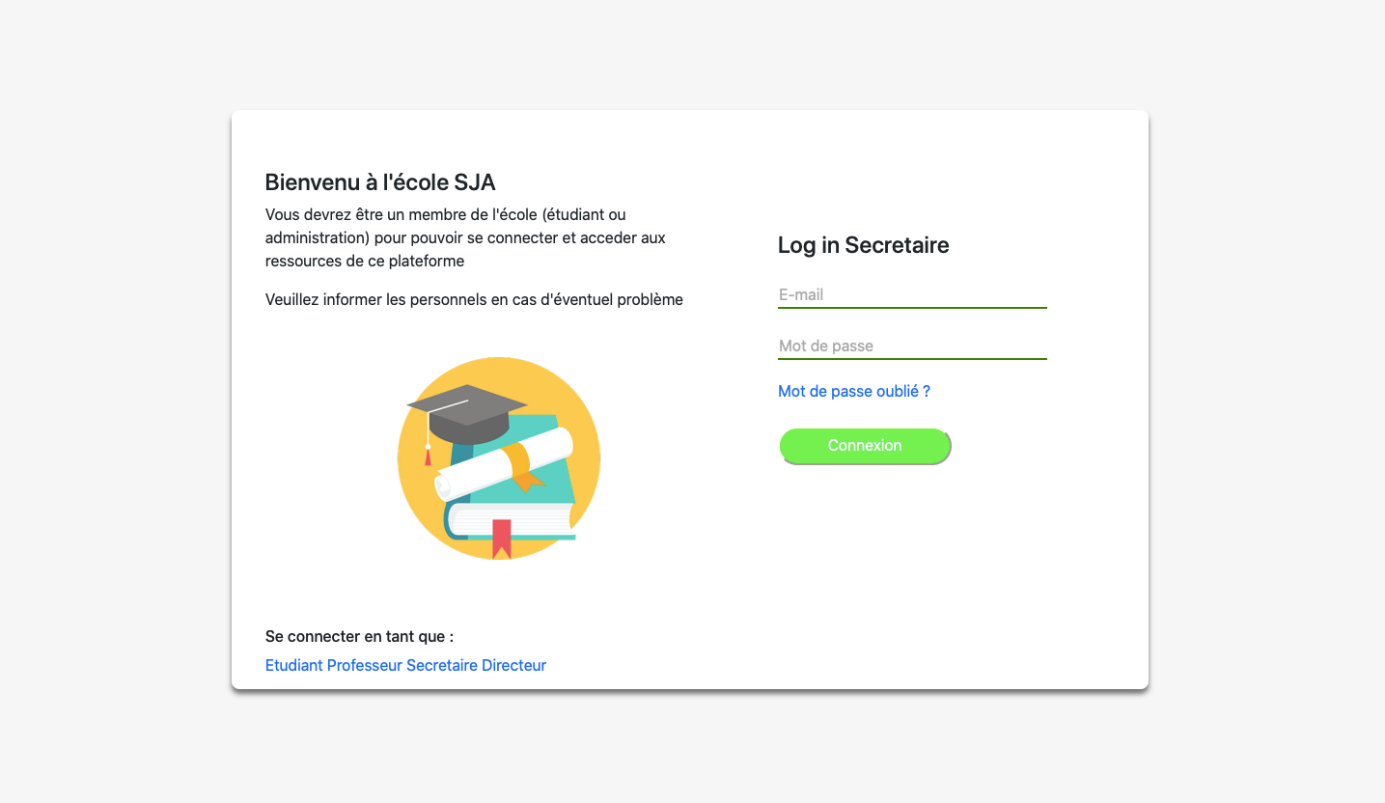
EtudiantsLog.jsp



ProfLog.jsp

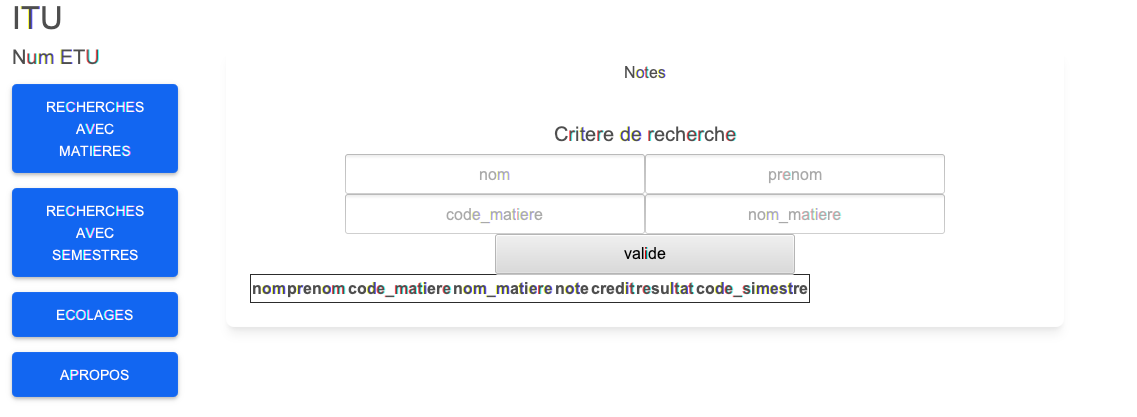


SecretaireLog.jsp

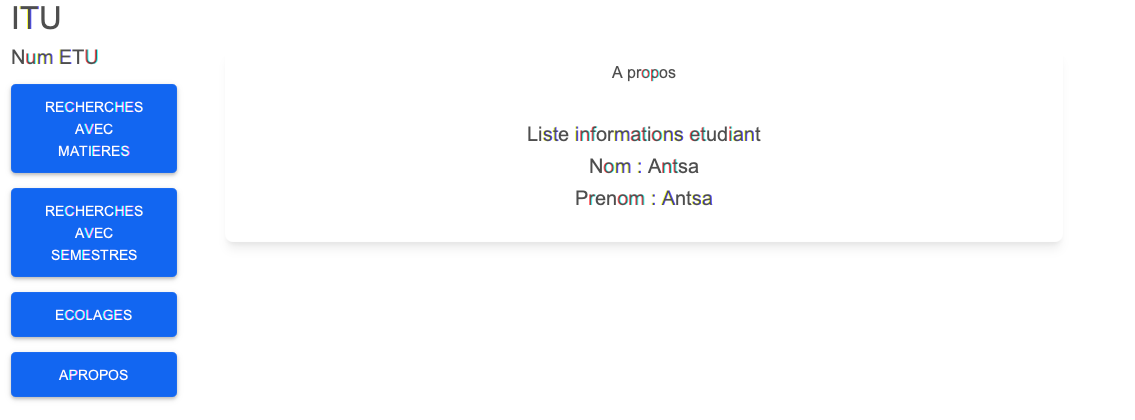


1. Étudiants :

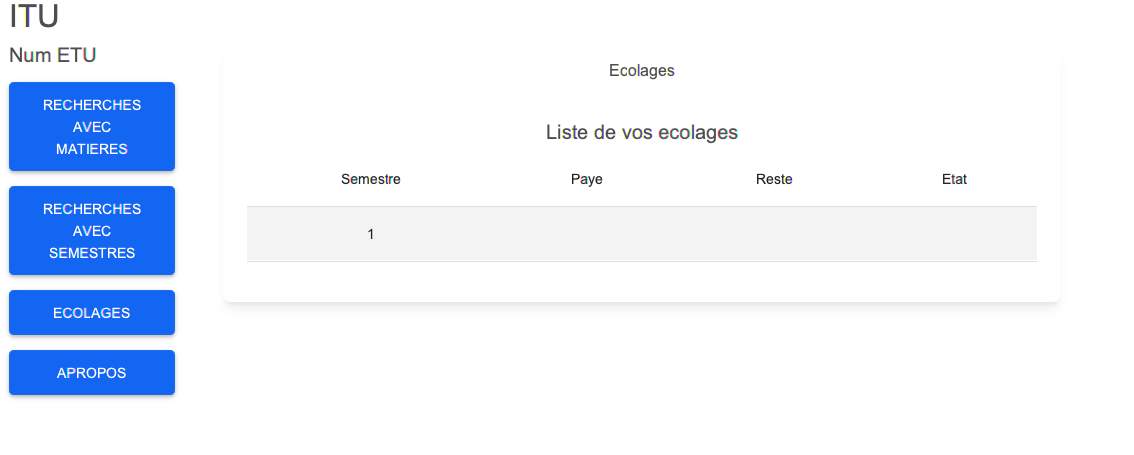
* Notes.jsp

****

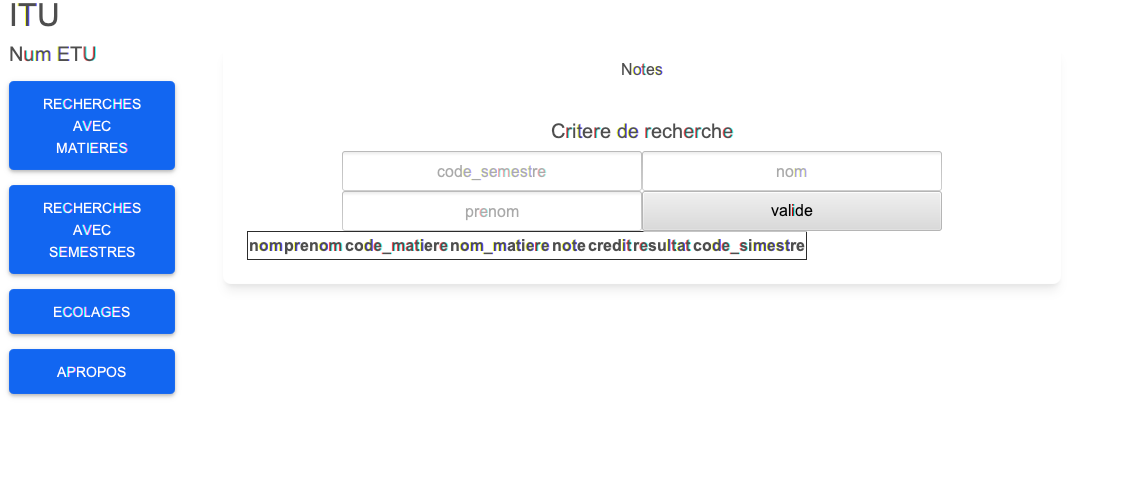
* Apropos.jsp



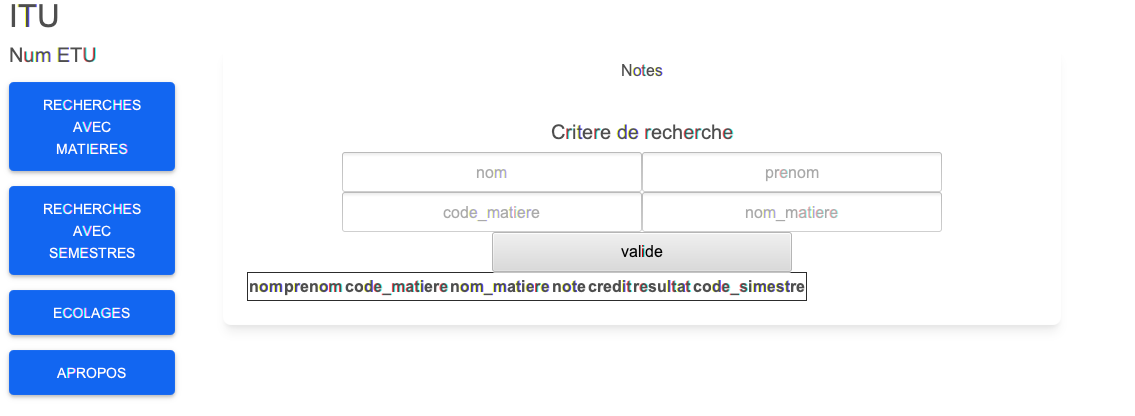
* Ecolages.jsp



* RechercheSem.jsp

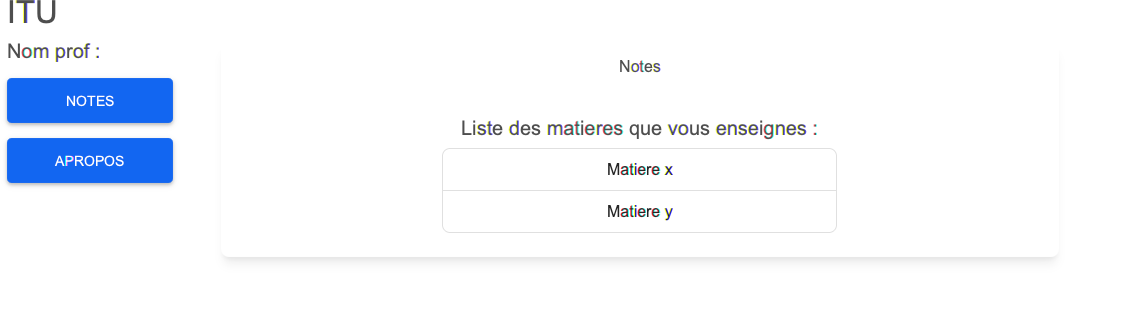


* RechercheMat.jp

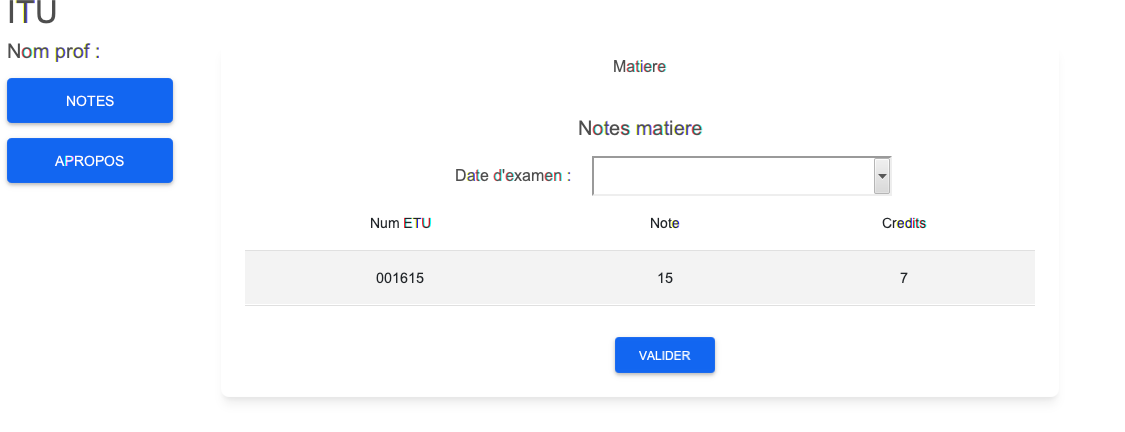


1. Professeurs :

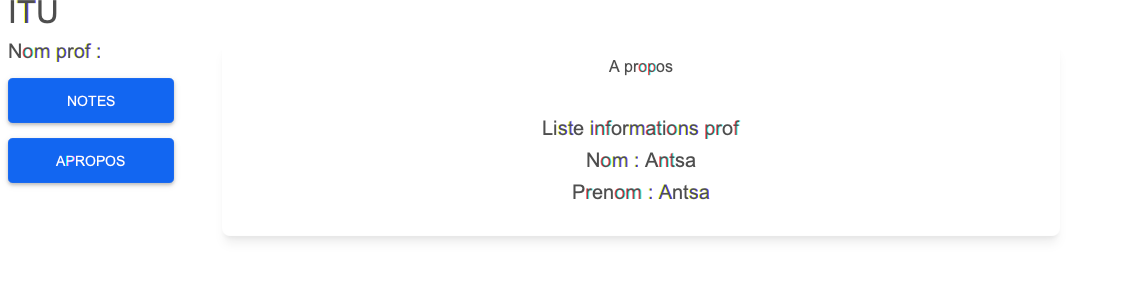
-profs.jsp



* Matieres.jsp

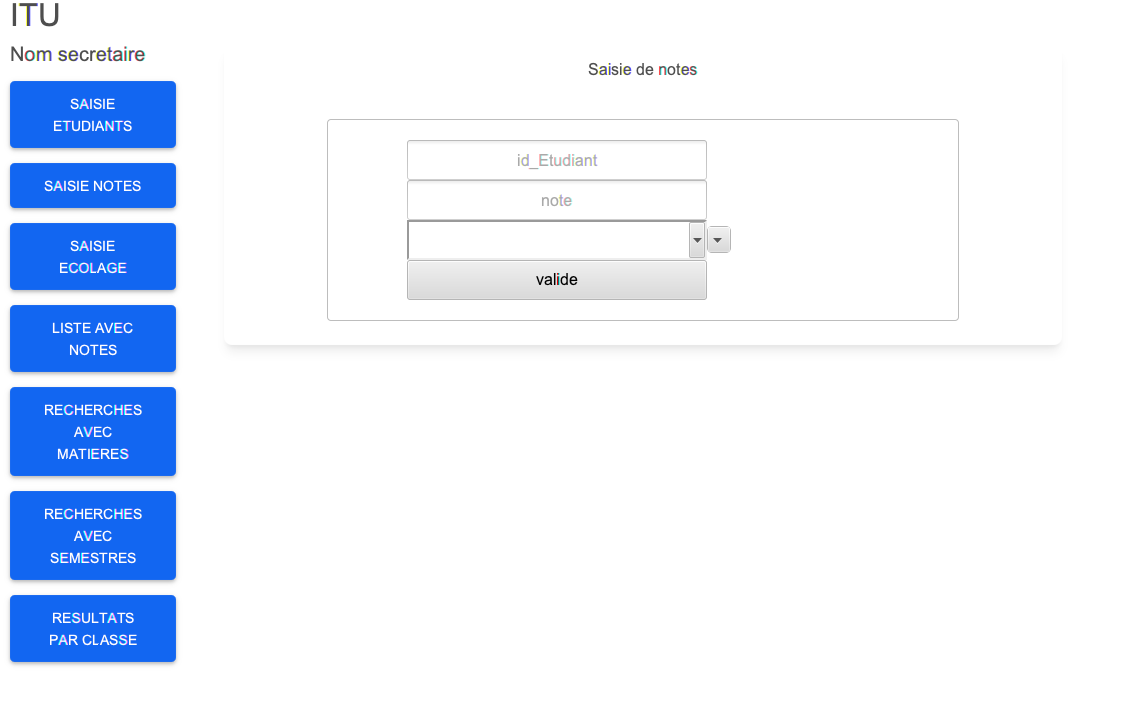


* AproposProfs.jsp

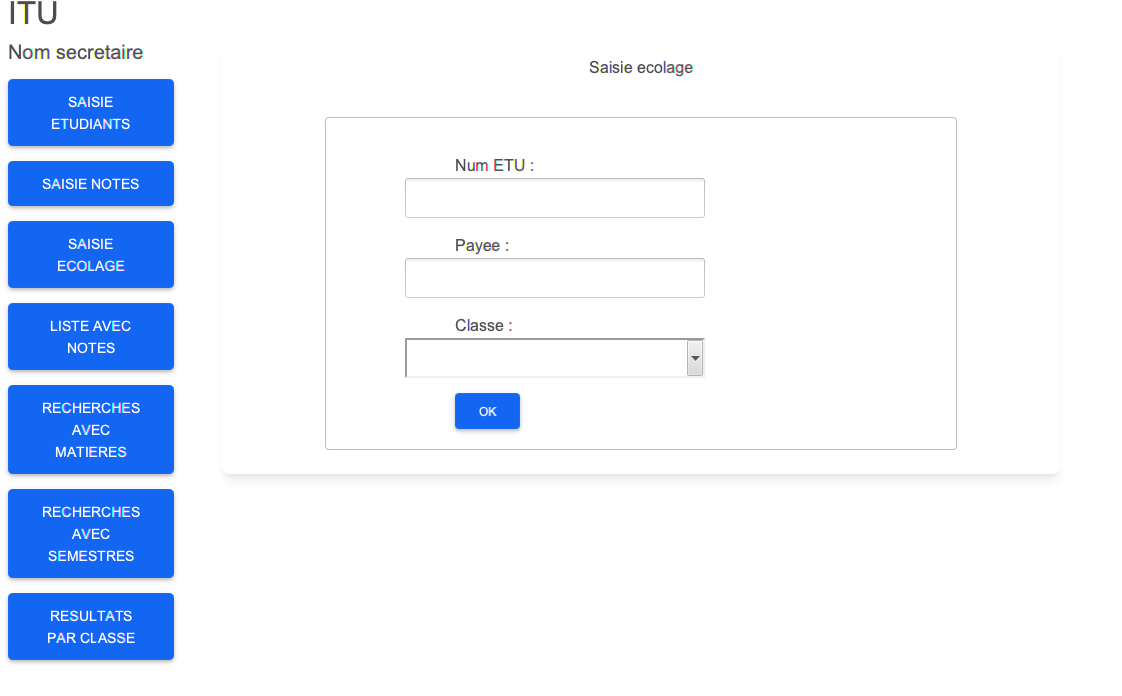


1. Secrétaire

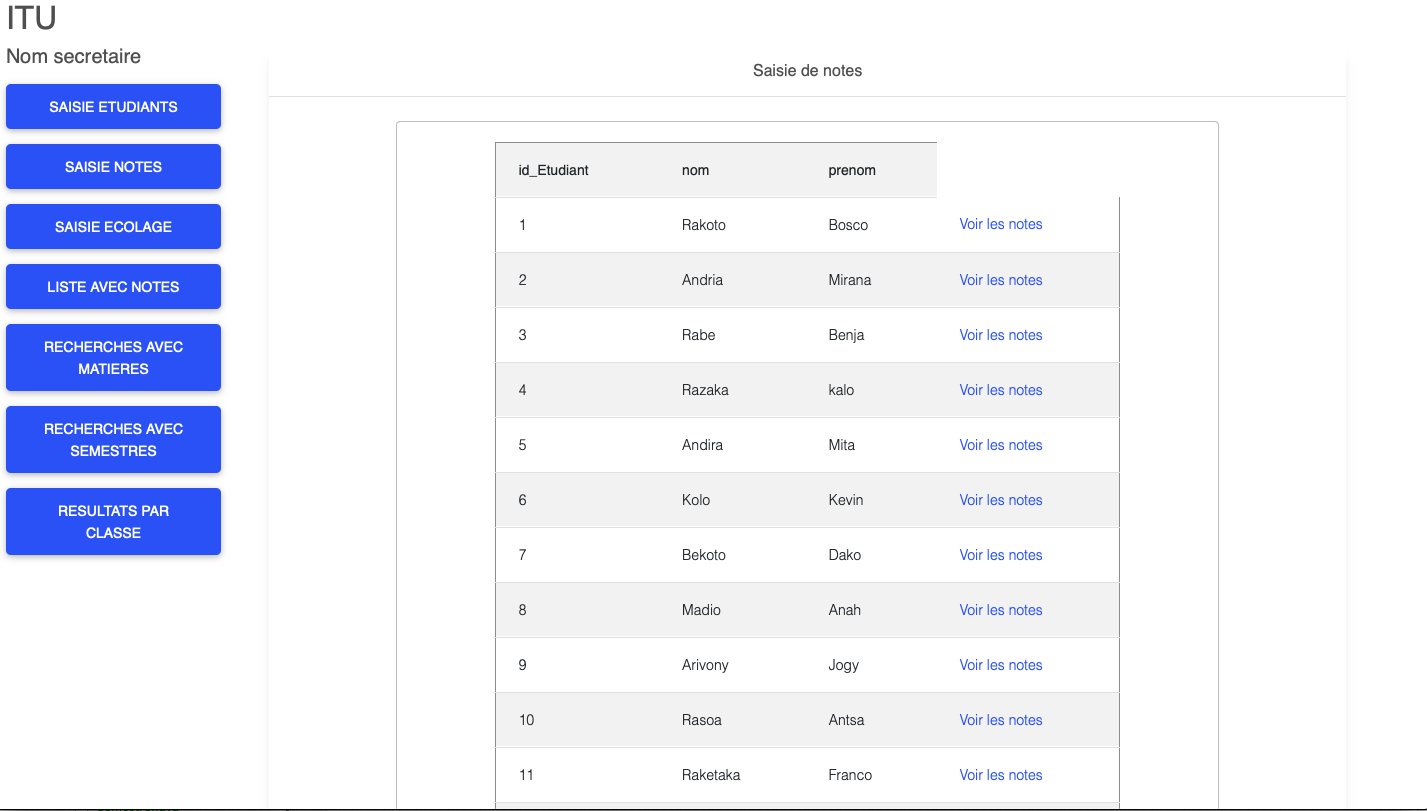
* SecretaireNote.jsp



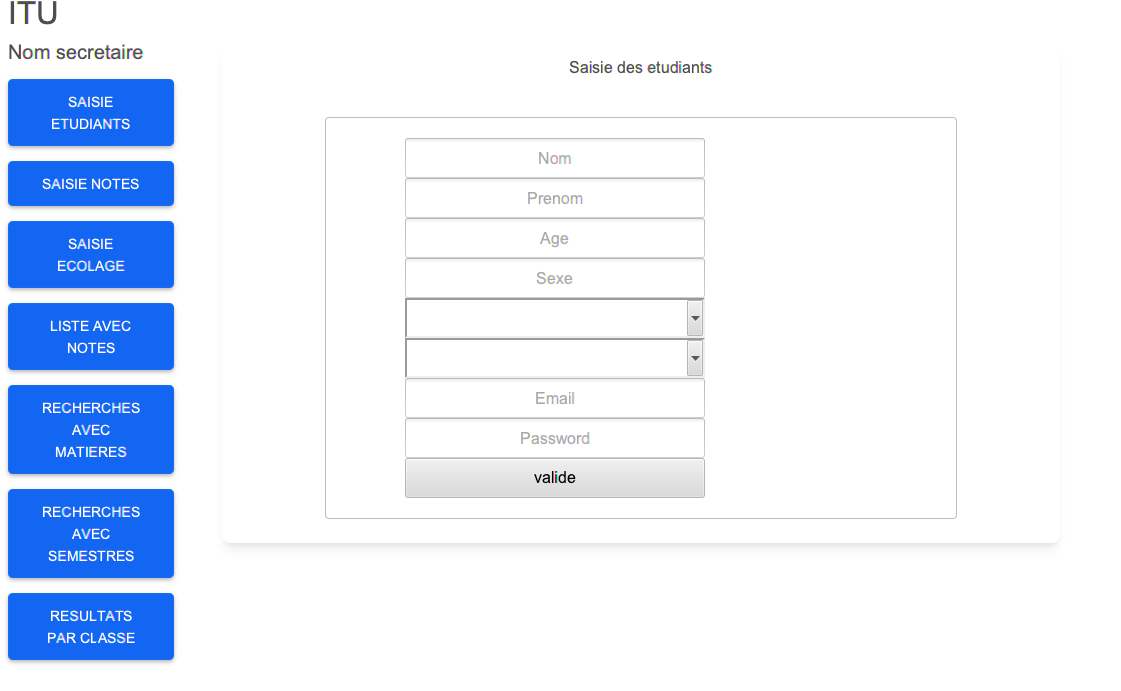
* SecretaireEco.jsp



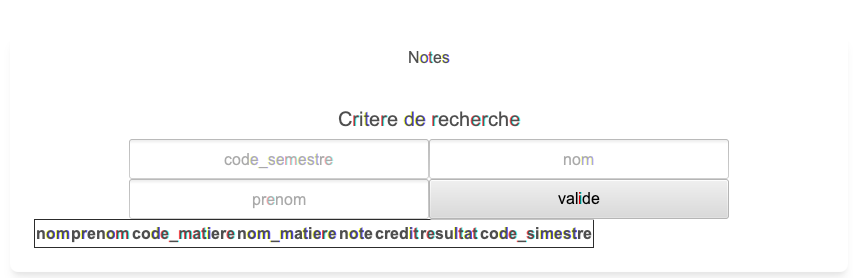
* Liste.jsp



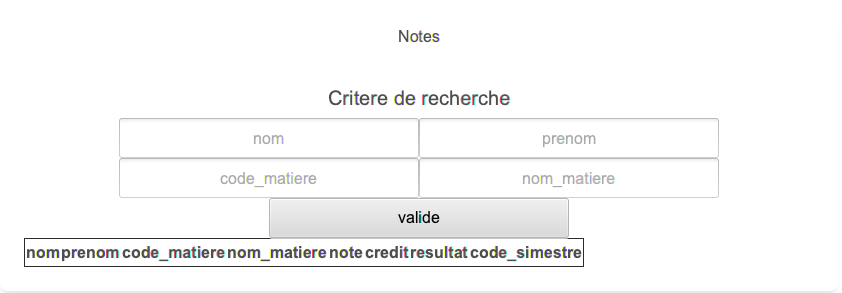
* SecretaireEtu.jsp



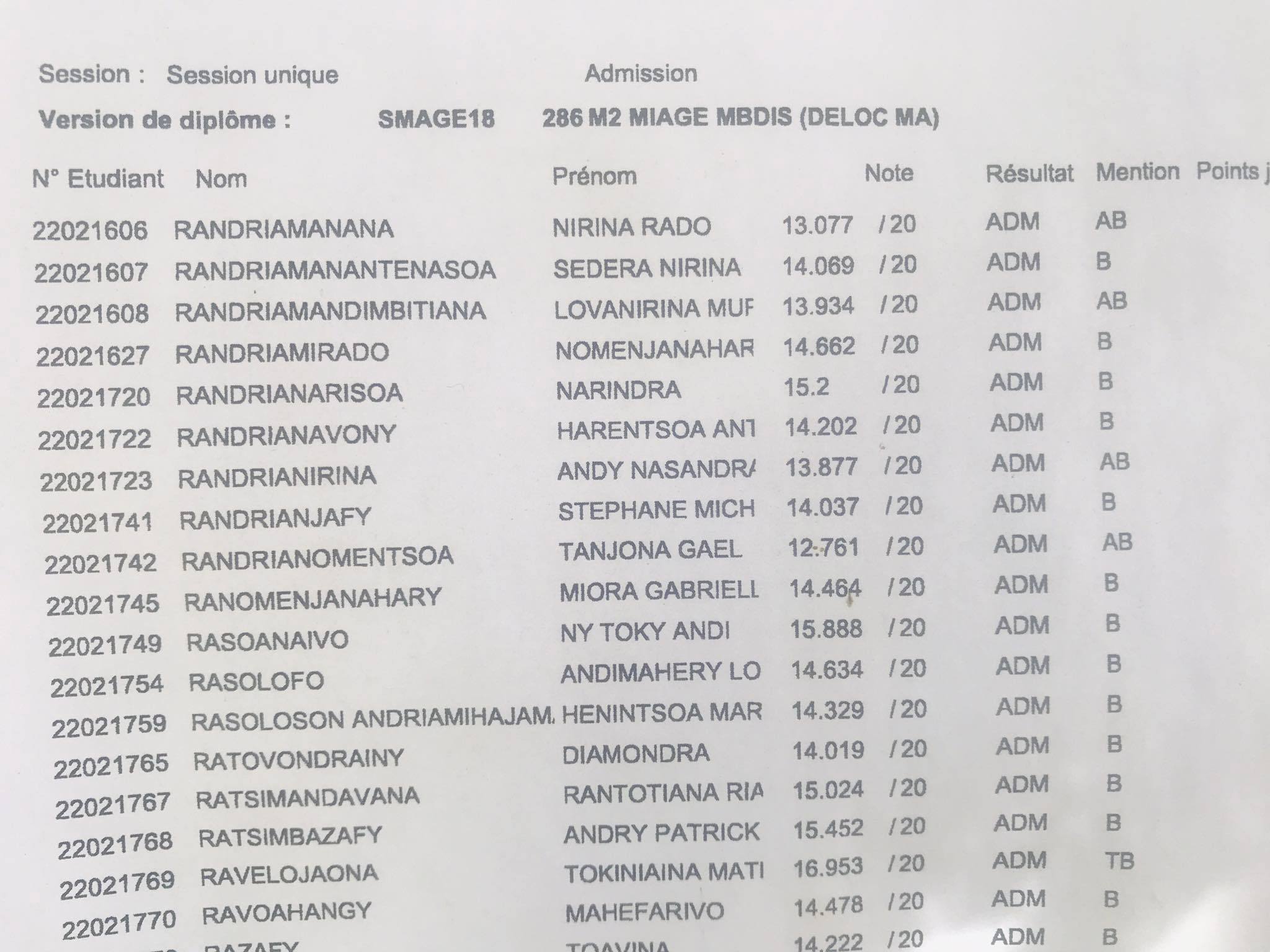
* SecretaireSem.jsp



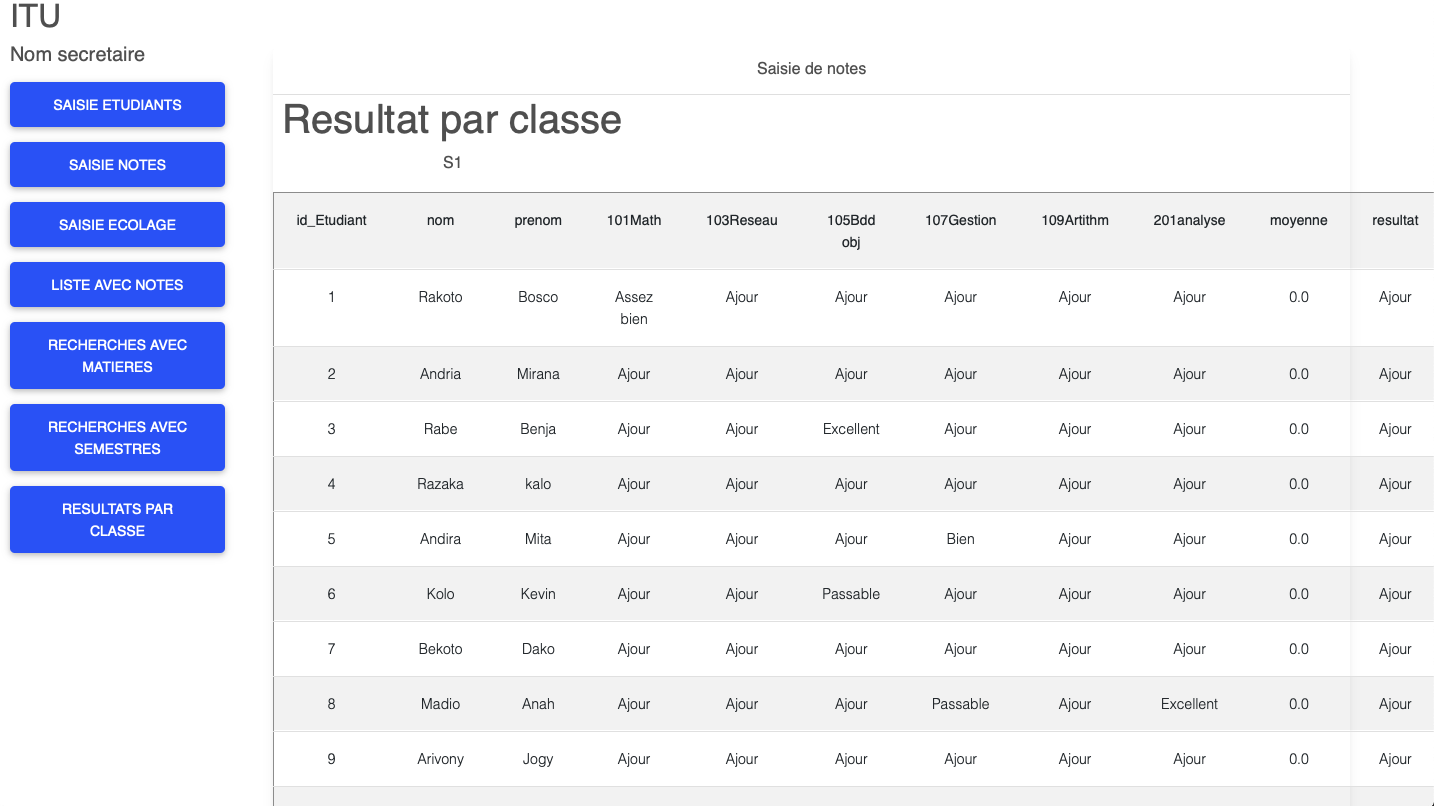
* SecretaireMatieres.jsp



* SecretaireResultat.jsp

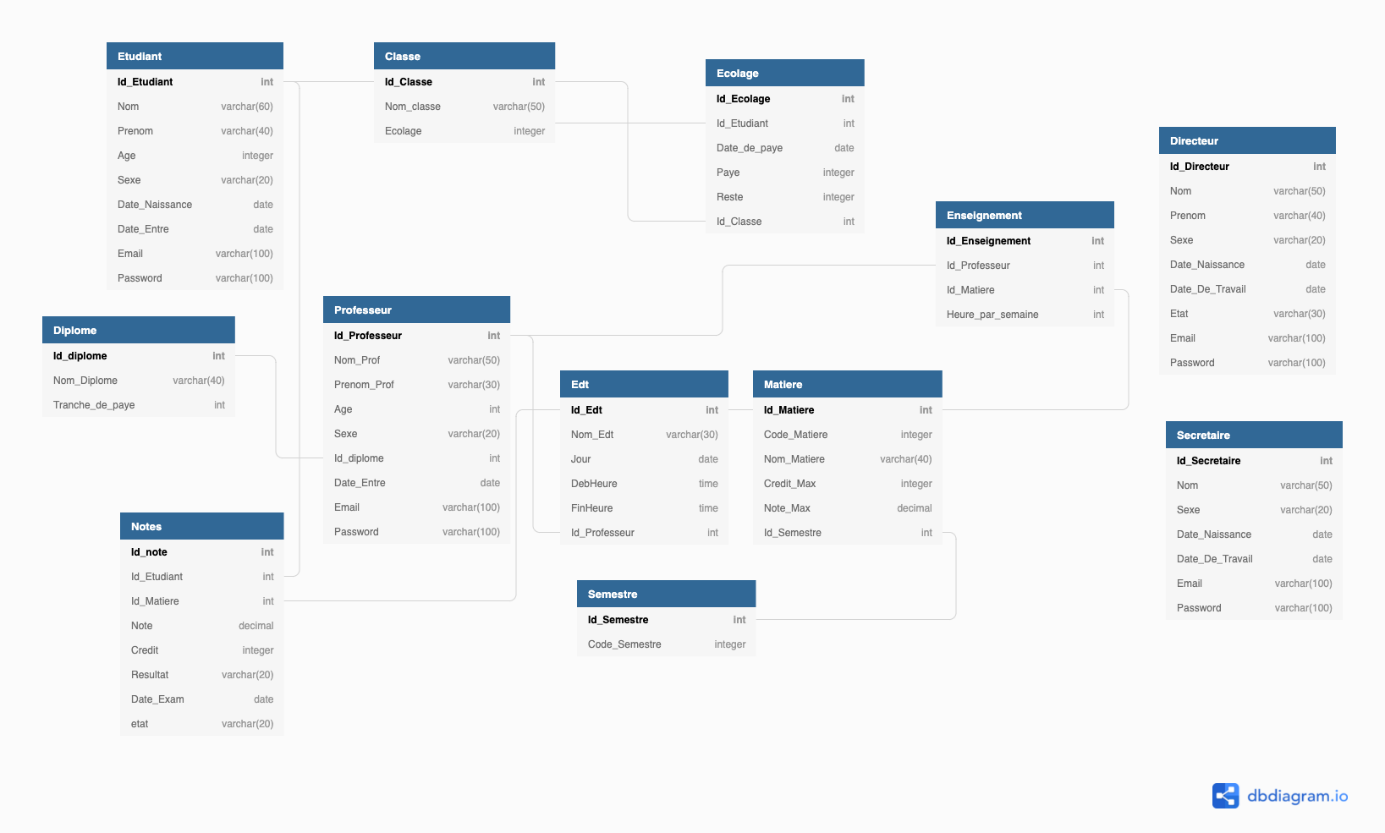


N.B : Le design reste le même mais ce qui sera dans le contenu ressemblera a ce résultat ci-dessus.

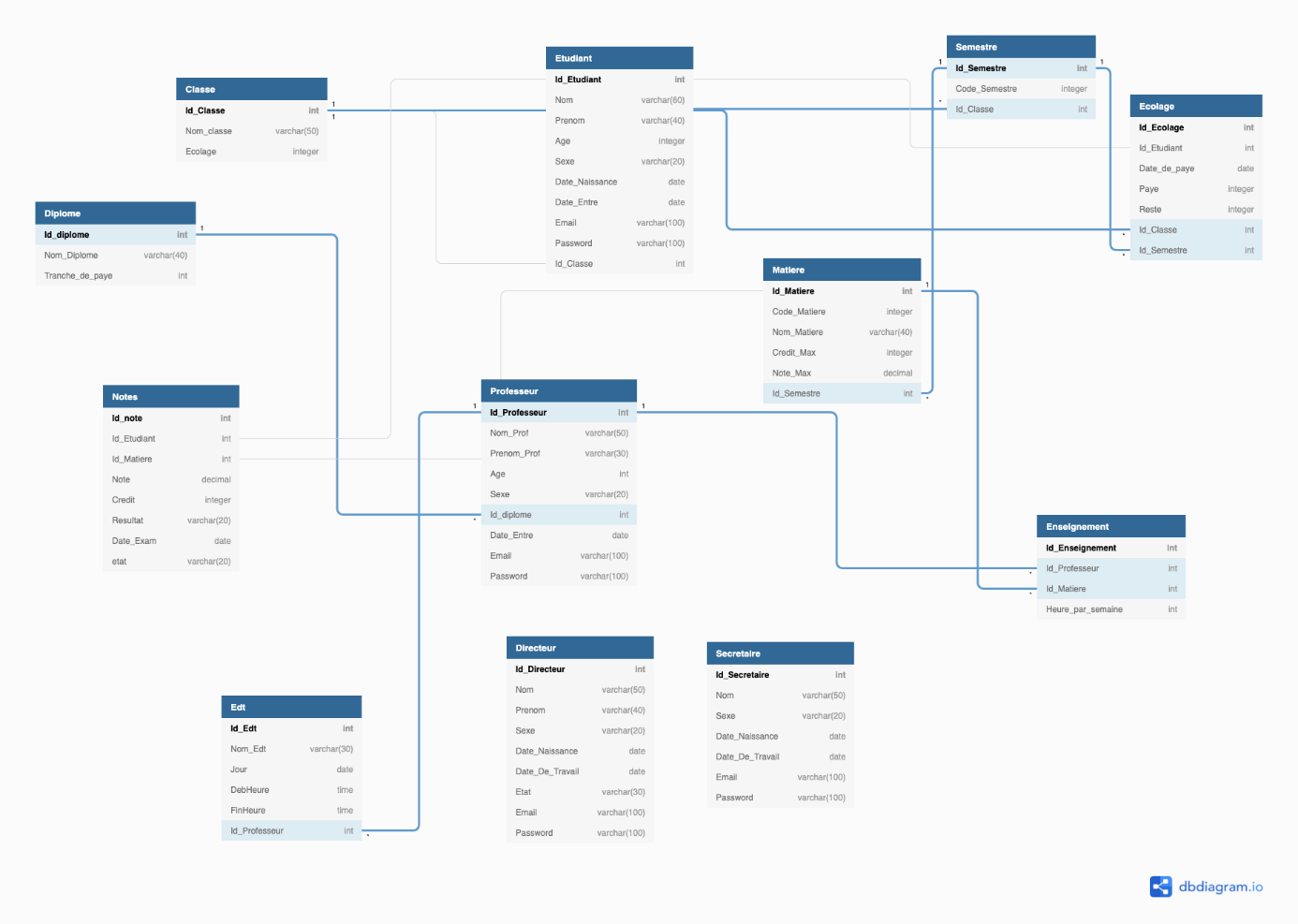


**B) Base de données :**

Voici le MCD du base du logiciels de gestion d’écoles



Après aléas



**Voici les vue à créer :**

**--a)Views Semestriel (Jointure Matiere/Semestre)**

Create or replace view Semestriel as Select Matiere.Id\_Matiere,Matiere.Code\_Matiere,Matiere.Nom\_Matiere,Matiere.Credit\_Max,Matiere.Note\_Max,Matiere.Id\_Semestre,Semestre.Code\_Semestre from Matiere join Semestre on Matiere.Id\_Semestre=Semestre.Id\_Semestre;

**--b)Views Notee (Jointure Note/Matiere)**

Create or replace view Notee as Select Notes.Id\_Note,Notes.Id\_Etudiant,Notes.Note,notes.Credit,Notes.Resultat,Notes.Date\_Exam,Semestriel.Code\_Matiere,Semestriel.Nom\_Matiere,Semestriel.Credit\_Max,Semestriel.Note\_Max,Semestriel.Code\_Semestre from Notes join Semestriel on Notes.Id\_Matiere=Semestriel.Id\_Matiere;

**--c) views Noteeleves (Jointure Etudiant/Note)**

Create or replace view Noteeleves as Select Notee.Id\_Etudiant,Notee.Note,notee.Credit,Notee.Resultat,Notee.Date\_Exam,Notee.Code\_Matiere,Notee.Nom\_Matiere,Notee.Credit\_Max,Notee.Note\_Max,Notee.Code\_Semestre,Etudiant.Nom,Etudiant.prenom from Notee join Etudiant on Notee.Id\_Etudiant=Etudiant.Id\_Etudiant;

**--d) views Sem\_mat ( Jointuer semestre et par matiere)**

Create or replace view Sem\_mat as Select count(\*)as Nombre, Code\_Matiere,Code\_Semestre from Noteleeves group by Code\_Matiere,Code\_Semestre;

**--e)views Paiements (Jointure Ecolage/Etudiant)**

Create or replace view Paiements as Select Ecolage.Id\_Etudiant,Ecolage.Date\_de\_paye,Ecolage.Paye,Ecolage.Reste,Etudiant.nom,Etudiant.Prenom,Etudiant.age from Ecolage join Etudiant on Ecolage.Id\_Etudiant= Etudiant.Id\_Etudiant;

**--f) views ecolages**

Create view ecolages as Select Paiements.Id\_Etudiant,Paiements.Paye,Paiements.Nom,Paiements.Prenom from Paiements order by Id\_Etudiant asc;

**--g) views moyenness**

Create view moyenness as Select Id\_Etudiant,avg(Note)as moyenne from Noteeleves group by Id\_Etudiant,Code\_Semestre order by Id\_Etudiant asc ;

**2)Métiers :**

Les classe à créer sont :

* Ecolage.java

Fonction :

1. String[][] liste\_Ecolage\_Par\_Niveau()
2. Void insertEcolage(Ecolage ecolage)
3. Double getTenPerCent(int idClasse)
4. Boolean checkSemestresIsPayed(int idSemestres, int idetudiant)
5. Double getPaidEco(int idEtudiant,int idClasse)
6. Double getRestEco(int Etudiant,int idClasse)

* Etudiant.java

Fonction :

1. ajout\_Etudiant(String[] etude)
2. String traitementAjout(String[] etude)
3. String[][] listerEtudiant()
4. Boolean checkIsthere(int IdETudiant)
5. Boolean checkIfRedoublant(int idEtudiant)

* Note.java

Fonction :

1. String[][] resultatNoteParSemetre(String code\_Semestre)
2. String[] titreParSemestre(String code\_Semestre)
3. String representationResultat(String noteMatiere,String credit)
4. String calculeSomme(String[][] list,int indice)
5. String calculeMoyenne(String[][] list,int indiceN,int indiceM,String credit)
6. void validationNote(String id\_Etudiant,String code\_matiere)
7. ajoutNote(String id\_Etudiant, String note,String date\_Exam,String id\_matiere)
8. String traitement\_Resultat(String note,String id\_matiere)
9. String[][] getId\_Matiere(String code\_matiere)
10. String[][] creditMatiere(String id\_matiere)
11. String[][] listeNotePar\_Semestre\_Etudiant\_Matiere(String code\_Semestre,String id\_Etudiant,String code\_matiere)
12. String[][] listeNoteParSemtre(String code\_Semestre,String id\_Etudiant)
13. String[][] filtreListeNote(String[] contexte,String[] value,String[] type)
14. String analyseFiltre(String[] contexte,String[] values,String table,String[] type)

* Semestre.java

Fonction :

1. String[][] ListeSemestre()
2. String[][] listeMatiere()
3. String[][] listeMatiereParSemestre(String code\_Semestre)
4. Int getInfoSemestre(int idSemetres)

* Classe.java

Fonction :

1. Classe getInfoClasse(int idClasse)
2. Double getEcolageClasse(int idClasse)

* Matiere.java

Fonction :

1. Matiere getInfoMatiere(int idMatiere)