**Rapport De Mini Proje**

<<Système de gestion d’école en JAVA >>

Realiser par :

Sameh Chouaieb

Ahmed Boujlida

## **1. Introduction**

Dans un contexte éducatif en constante évolution, la gestion efficace des données liées aux étudiants, aux enseignants et au personnel administratif est devenue une nécessité pour les établissements scolaires. Ce projet s’inscrit dans cette logique en proposant le développement d’un **système de gestion d’école** complet, conçu pour automatiser et centraliser les principales opérations de l’institution.

L’objectif principal de ce système est de fournir une interface conviviale et sécurisée permettant aux différents utilisateurs — **administrateurs**, **enseignants** et **étudiants** — d’accéder à leurs informations et de gérer les données selon leurs rôles respectifs. Les fonctionnalités comprennent la gestion des étudiants et des enseignants, l’affectation des matières, ainsi que la consultation des informations essentielles.

Le projet a été développé en **Java**, en utilisant la bibliothèque **Swing** pour la conception des interfaces graphiques, et une base de données **MySQL** pour le stockage et la gestion des données. Ce choix technologique permet d’assurer une solution robuste, modulaire et facilement maintenable.

Dans ce rapport, nous présentons l’ensemble du processus de conception et de réalisation du système. Il comprend la modélisation UML, la structure de la base de données, la description des interfaces, ainsi que des extraits de code et des captures d’écran illustrant le fonctionnement du système.

**2. Cahier des charges**

**2.1 Objectif du système**

Le système de gestion d’école a pour but de centraliser et simplifier la gestion des données administratives et pédagogiques au sein d’un établissement. Il permet de gérer les informations des étudiants, des enseignants, du personnel administratif et des matières enseignées. Il offre également des interfaces personnalisées en fonction du profil de l'utilisateur connecté.

**2.2 Types d'utilisateurs**

Le système distingue **trois types principaux d'utilisateurs**, chacun ayant un accès restreint à des fonctionnalités spécifiques :

**a) Administrateur / Personnel**

* Gère les données des étudiants et des enseignants.
* Peut ajouter, modifier, supprimer des enregistrements.
* Peut affecter des matières aux enseignants.

**b) Enseignant**

* Peut consulter la liste des étudiants.
* Peut consulter la liste des matières qu’il enseigne.

**c) Étudiant**

* Peut consulter la liste des matières.
* Peut consulter la liste des enseignants.

Chaque utilisateur doit s’authentifier avec un **nom d’utilisateur** et un **mot de passe**. Une fois connecté, il est redirigé vers une interface correspondant à son rôle.

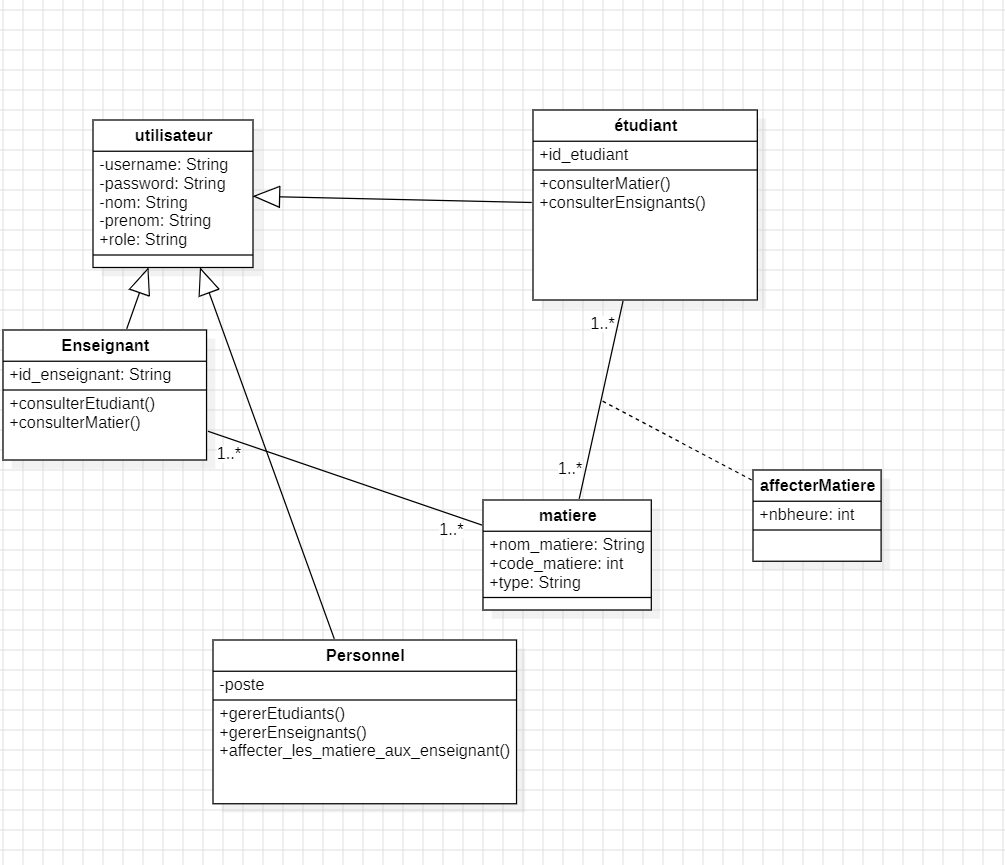
***Conception UML :***

Diagramme de cas d’utilisation

A diagram of a diagram

AI-generated content may be incorrect.

Diagramme de classe :



**Modèle de base de données:**

Utilisateur(log , pass ,nom,prenom ,user\_role)

Enseignant(log, pass,nom,prenom,specialité)

Etudiant(log ,pass,nom,prenom, specialité)

Personnel(log , pass ,nom , prenom ,poste )

Matiere ( id, nom , type )

affectationmatens (#id\_enseignant, #id\_matiere, nbheure)

LES REQUETTES SQL :

create database école;

CREATE TABLE utilisateur (

log VARCHAR(50) UNIQUE NOT NULL PRIMARY KEY,

pass VARCHAR(255) NOT NULL,

user\_role ENUM('admin', 'enseignant', 'etudiant') NOT NULL

,nom VARCHAR(100) NOT NULL,

prenom VARCHAR(100) NOT NULL

);

CREATE TABLE etudiant (

log INT PRIMARY KEY,

nom VARCHAR(100) NOT NULL,

pass VARCHAR(100) ,

prenom VARCHAR(100) NOT NULL

,specialité VARCHAR(100),

niveau INT

);

CREATE TABLE enseignant (

log INT PRIMARY KEY,

nom VARCHAR(100) NOT NULL,

prenom VARCHAR(100) NOT NULL,

specialite VARCHAR(100),

pass varchar(100)

);

CREATE TABLE personnel (

log INT PRIMARY KEY,

nom VARCHAR(100) NOT NULL,

prenom VARCHAR(100) NOT NULL,

poste VARCHAR(100) );

CREATE TABLE matiere (

id INT AUTO\_INCREMENT PRIMARY KEY,

nom VARCHAR(100) NOT NULL,

type VARCHAR(100) NOT NULL

);

CREATE TABLE affectationmatens (

id\_enseignant INT NOT NULL,

id\_matiere INT NOT NULL,

FOREIGN KEY (id\_enseignant) REFERENCES enseignant(log),

FOREIGN KEY (id\_matiere) REFERENCES matiere(id)

);

**Page de connexion (Login)**

La page de **connexion** est la **porte d'entrée du système**. Elle permet à tous les utilisateurs de s’authentifier via un **nom d’utilisateur** et un **mot de passe**. En fonction des informations fournies et du **rôle** associé dans la base de données (étudiant, enseignant ou personnel/administrateur), l’utilisateur est redirigé automatiquement vers l’interface qui lui correspond.

**Composants de l’interface Login :**

* Champ **Nom d’utilisateur** : zone de texte pour saisir le nom de connexion.
* Champ **Mot de passe** : zone de texte sécurisée pour entrer le mot de passe.
* Bouton **Connexion** : déclenche le processus de vérification des identifiants.
* Message d’erreur : affiché en cas d’identifiants incorrects ou d’accès refusé.

**Fonctionnement :**

1. L’utilisateur saisit ses identifiants.
2. Le système vérifie l’existence d’un utilisateur dans la table utilisateur avec :
   * le **nom\_utilisateur** saisi,
   * le **mot\_de\_passe** (chiffré si implémenté),
3. Si la connexion est réussie :
   * Si role = 'personnel', il est redirigé vers l’**interface administrateur**.
   * Si role = 'enseignant', il accède à l’**interface enseignant**.
   * Si role = 'etudiant', il accède à l’**interface étudiant**.

A screenshot of a login screen

AI-generated content may be incorrect.

1. Sinon, un **message d'erreur** s’affiche pour informer que la connexion a échoué.

A screenshot of a login screen

AI-generated content may be incorrect.

**6. Interface Administrateur (Personnel)**

L'interface **Administrateur** est accessible uniquement aux utilisateurs ayant le rôle **"personnel"** dans le système. Elle constitue la **zone de gestion principale** de l'application et permet d'effectuer toutes les opérations relatives à la gestion des données de l’école.

**6.1 Objectif de l’interface**

Permettre au personnel administratif de :

* Gérer les informations des **étudiants**.
* Gérer les informations des **enseignants**.
* **Affecter les matières** aux enseignants.