

# RAPPORT DE PROJET

# JEE

*Réalisation d'une application web de gestion de scolarité*

**Yasmina BOUZIANE**

**Sara GUENDOUZ,**

**Maxime PAVIET,**

**Anis SOUSSA**

30/11/2024

## Sommaire

- I. Introduction**
- II. Conception de l'application (Jakarta)**
  - Architecture Générale
  - Diagramme de classes
  - Base de données
- III. Fonctionnalités de l'application**
  - Ouverture de l'application
  - Espace enseignants
  - Espace étudiants
  - Espace administrateur
  - Envoie de mail
- IV. Conception de l'application (Springboot)**
  - Architecture Générale
  - Diagramme de classes
  - Base de données
- V. Conclusion**
  - Bilan du projet
  - Perspectives d'amélioration
  - Retour d'expérience

## Introduction

L'objectif principal du projet JEE est de développer une application web dynamique permettant de simplifier et de centraliser la gestion des informations académiques au sein d'un établissement d'enseignement. Ce projet vise à offrir une solution moderne et intuitive pour répondre aux besoins des trois principales parties prenantes : les étudiants, les enseignants, et les administrateurs. L'application permettra de gérer efficacement les données liées aux étudiants, enseignants, cours, inscriptions, et résultats, tout en garantissant une expérience utilisateur optimisée.

L'application web est conçue en utilisant les technologies du framework Java EE, avec une architecture MVC (Modèle-Vue-Contrôleur). Il exploite Hibernate pour le mappage objet-relationnel et MySQL pour la gestion de la base de données.

Parmi ses fonctionnalités clés, l'application offre la gestion complète des utilisateurs et des cours, l'attribution des enseignants aux cours, l'inscription des étudiants, ainsi qu'un système de saisie et de consultation des notes. En outre, un module de notification par e-mail assure que les utilisateurs restent informés des mises à jour importantes telles que les inscriptions, les notes et les modifications de cours. Cette solution se distingue également par une sécurité renforcée grâce à un système d'authentification et de gestion des rôles.

## Conception de l'application (Jakarta) : (Branche jakarta-main)

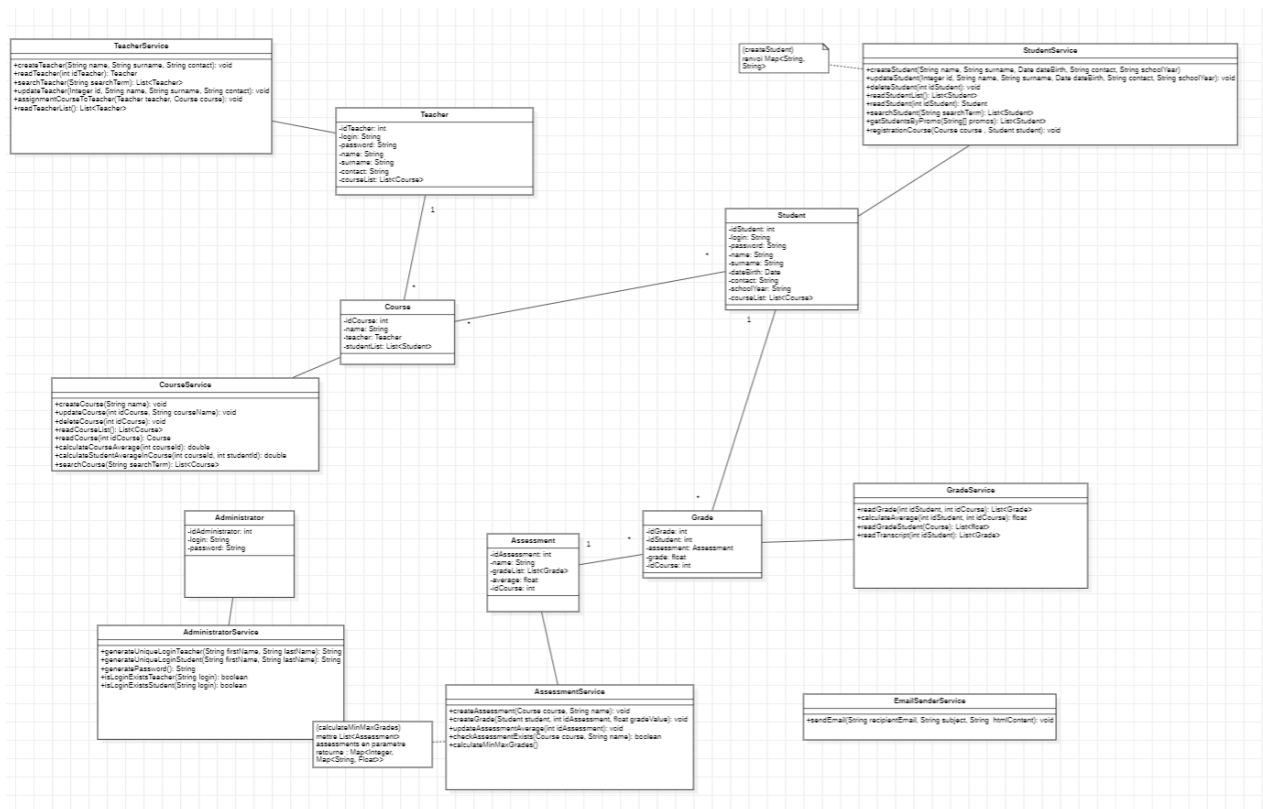
### Architecture Générale :

#### 1. Les principaux packages Java utilisés :

- **Package `controllers`** : Contient les servlets responsables de gérer les interactions entre l'utilisateur et le système. Chaque servlet correspond à une fonctionnalité spécifique comme par exemple `StudentPageAdminServlet` ou `AddAssessmentTeacherServlet`. Ces classes contrôlent les requêtes HTTP et orchestrent la logique métier.
- **Package `models`** : Comprend les classes représentant les entités principales de l'application :
  - **`Student`** : représente un étudiant
  - **`Teacher`** : représente un enseignant,
  - **`Administrator`** : représente un administrateur,
  - **`Course`** : décrit un cours
  - **`Assessment`** : représente une évaluation pour un cours.
  - **`Grade`** : associer un étudiant à une évaluation avec une note
- **Package `services`**: Contient les classes responsables de la logique métier et de la communication avec la base de données :
  - **`StudentService`** : gère les opérations sur les étudiants (création, mise à jour, suppression).
  - **`TeacherService`** : gère les opérations sur les enseignants.
  - **`CourseService`** : gère les opérations liées aux cours.
  - **`AssessmentService`** : gère les évaluations.
  - **`GradeService`**: gère les notes.
  - **`AdministratorService`**: gère les génération et vérification des login/mots de passe.
  - **`EmailSenderService`** : gère le système de notifications par e-mail aux étudiants.

---

#### 2. Diagramme de classe :



### 3. Base de données :

La base de données est composée des tables suivantes :

- **administrator** : Contient les informations des administrateurs :
  - **idAdministrator** (INT, clé primaire)
  - **login** (VARCHAR, unique)
  - **password** (VARCHAR)
- **student** : Stocke les informations des étudiants :
  - **idStudent** (INT, clé primaire)
  - **name** (VARCHAR)
  - **surname** (VARCHAR)
  - **dateBirth** (DATE)
  - **contact** (VARCHAR)
  - **schoolYear** (VARCHAR)
- **teacher** : Contient les informations des enseignants :
  - **idTeacher** (INT, clé primaire)
  - **name** (VARCHAR)
  - **surname** (VARCHAR)
  - **contact** (VARCHAR)

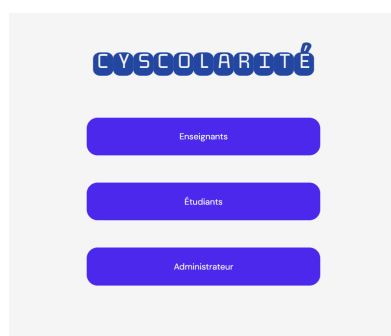
- **course** : Représente les cours avec les colonnes suivantes :
  - **idCourse** (INT, clé primaire)
  - **name** (VARCHAR)
  - **teacher\_id** (INT, clé étrangère vers **teacher**)
- **student\_course** : Table intermédiaire pour les relations "étudiant-cours" :
  - **student\_id** (INT, clé étrangère vers **student**)
  - **course\_id** (INT, clé étrangère vers **course**)
- **assessment** : Stocke les informations sur les évaluations :
  - **idAssessment** (INT, clé primaire)
  - **name** (VARCHAR)
  - **idCourse** (INT, clé étrangère vers **course**)
- **grade** : Associe les notes aux étudiants et aux évaluations :
  - **idGrade** (INT, clé primaire)
  - **idStudent** (INT, clé étrangère vers **student**)
  - **idAssessment** (INT, clé étrangère vers **assessment**)
  - **grade** (FLOAT)

Ces structures relationnelles assurent la cohérence des données et facilitent leur manipulation via Hibernate.

## Fonctionnalités de l'application

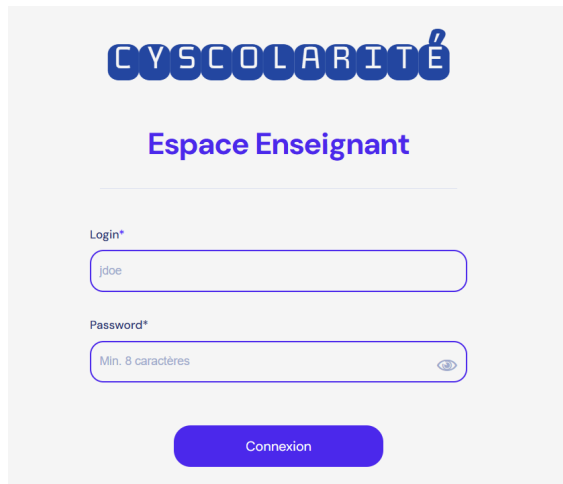
### Ouverture de l'application:

Lors de l'ouverture de l'application, l'utilisateur sélectionne son rôle :



### Espace Enseignants:

En choisissant l'espace enseignants, l'utilisateur doit saisir son login et son mot de passe.



The screenshot shows the login interface for the 'CYSCOLARITÉ' platform, specifically for the 'Espace Enseignant' (Teacher Space). The header features the 'CYSCOLARITÉ' logo in a stylized blue font. Below the header, the title 'Espace Enseignant' is displayed in a bold, dark blue font. The login form consists of two input fields: 'Login\*' and 'Password\*'. The 'Login\*' field contains the text 'jdoe'. The 'Password\*' field has a placeholder 'Min. 8 caractères' and a toggle icon for visibility. A blue 'Connexion' button is positioned below the password field.

Si l'utilisateur est bien reconnu comme étant un enseignant, il accède à son profil :



The screenshot displays the 'Mon Profil' (My Profile) page. At the top left, there is a back arrow icon. The title 'Mon Profil' is centered at the top in a bold, dark blue font. Below the title, a box contains the user's details: 'Nom : Haddache.', 'Prénom : Mohamed', and 'Contact : MohamedHaddache@cy-tech.fr'. To the right of this box is a 'S' icon. Below the details box, the section 'Liste des cours :' is followed by a 'S' icon and a 'S'inscrire à un cours' button. A table lists the courses taught by the user.

Nom du cours	Nombre d'élèves	Moyenne
JEE	3	15,17
Architecture Réseau	2	Pas encore de notes

Il peut alors consulter ses informations, les différents cours qu'il enseigne ainsi que le nombre d'élèves dans chacune de ses matières et la moyenne. Il peut choisir de s'inscrire en tant qu'enseignant à de nouveaux cours :

## S'inscrire à un cours

Liste des cours existants :

Cours	Action
Cybersécurité	<input type="checkbox"/>
Anglais	<input checked="" type="checkbox"/>
IA	<input type="checkbox"/>
Test et Verification	<input type="checkbox"/>

S'inscrire

Il peut aussi obtenir plus d'informations sur un cours en cliquant sur la ligne correspondante dans le tableau. Voici un exemple pour JEE :

## JEE

Liste des évaluations :

Ajouter évaluation

Évaluation	Meilleure note	Pire note	Moyenne
TP1	17.0	13.5	15.1667

Une fois sur cette page l'enseignant peut créer une nouvelle évaluation pour cette matière par exemple "TP2" et renseigner les notes des élèves inscrit à ce cours :

## Ajouter une évaluation

Nom :

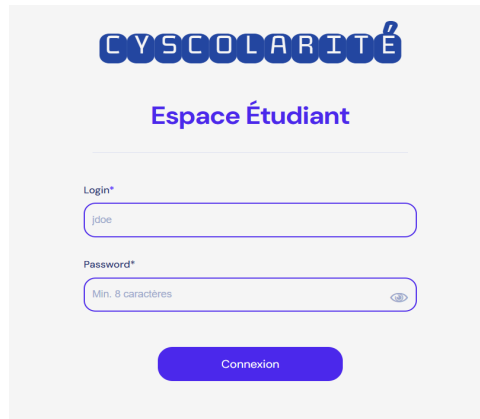
Nom	Prénom	Note
Soussa	Anis	<input type="text" value="15.5"/>
Paviet	Maxime	<input type="text" value="16.2"/>
Zeboudj	Nawel	<input type="text" value="14.6"/>

Ajouter



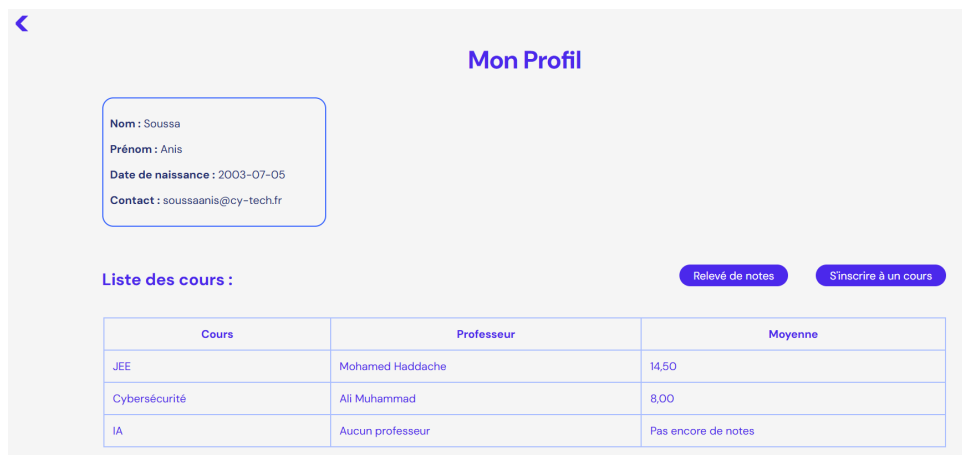
## Espace Etudiants:

En choisissant l'espace enseignants, l'utilisateur doit saisir son login et son mot de passe.



The screenshot shows the login interface for 'CYSCOLARITÉ Espace Étudiant'. At the top is the logo 'CYSCOLARITÉ' in blue. Below it, the title 'Espace Étudiant' is displayed in blue. The login form consists of two input fields: 'Login\*' with the text 'jdoe' and 'Password\*' with a hint 'Min. 8 caractères' and an eye icon. A blue 'Connexion' button is at the bottom.

Si l'utilisateur est bien reconnu comme étant un étudiant, il accède à son profil :



The screenshot shows the 'Mon Profil' page. At the top left is a back arrow. The title 'Mon Profil' is in blue. Below it is a profile card with the following information: Nom : Soussa, Prénom : Anis, Date de naissance : 2003-07-05, and Contact : soussaanis@cy-tech.fr. Below the card is the section 'Liste des cours :'. To the right of this section are two buttons: 'Relevé de notes' and 'S'inscrire à un cours'. Below these is a table with three columns: Cours, Professeur, and Moyenne.

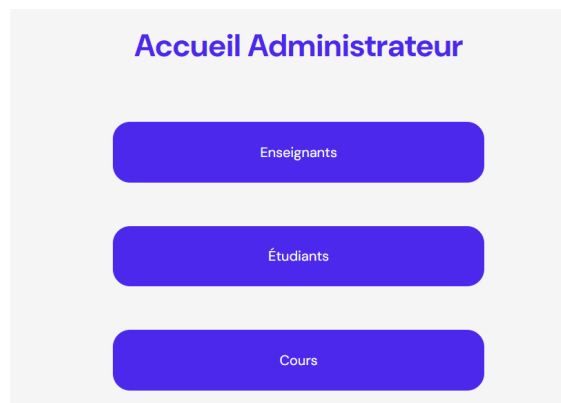
Cours	Professeur	Moyenne
JEE	Mohamed Haddache	14,50
Cybersécurité	Ali Muhammad	8,00
IA	Aucun professeur	Pas encore de notes

Il peut alors consulter ses informations, les différents cours auxquels il est inscrit ainsi que les professeurs enseignant ces cours et ses moyennes. Comme l'enseignant, l'étudiant peut s'inscrire à un cours (en cliquant sur le bouton cela ouvre une page similaire à celle de l'espace enseignant). Il peut également consulter son relevé de notes :

Relevé de notes	
IA	Pas encore de notes
Aucune évaluation pour ce cours.	
Cybersécurité	8,00
Rapport1	8,00
JEE	14,50
TP1	13,50
TP2	15,50

### Espace Administrateur :

Comme l'enseignant et l'étudiant, l'administrateur doit se connecter. Une fois qu'il est bien reconnu en tant qu'administrateur, un menu s'affiche :



Il peut alors choisir la section "Enseignants". S'ouvre alors une page avec la liste de tous les enseignants :

Page enseignants			
<input type="text" value="Recherche"/>		<input type="button" value="Ajouter Enseignant"/>	
Nom	Prénom	Contact	Action
Haddache	Mohamed	MohamedHaddache@cy-tech.fr	
Daniel	Valerie	ValerieDaniel@cy-tech.fr	
Mercadal	Julien	JulienMercadal@cy-tech.fr	
Moronenko	Natalia	NataliaMoronenko@cy-tech.fr	
Senoussi	Houcine	HoucineSenoussi@cy-tech.fr	
Muhammad	Ali	AliMuhammad@cy-tech.fr	
Forest	Jean-Paul	JeanForest@cy-tech.fr	

Il peut alors faire une recherche à l'aide de la barre de recherche dynamique qui permet d'effectuer une recherche sur le nom, le prénom ou le contact. Il peut aussi modifier les informations d'un enseignant grâce à la colonne "Action". Enfin il peut

créer un nouvel enseignant :

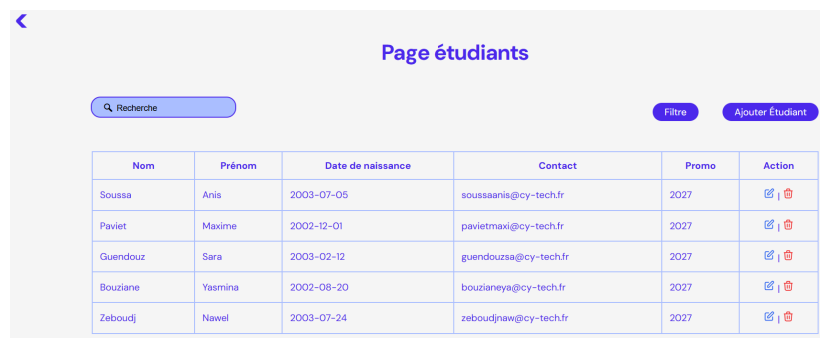


Ajouter enseignant

Nom :

Prénom :

S'il choisit la section "Etudiants". S'ouvre alors une page avec la liste de tous les étudiants :

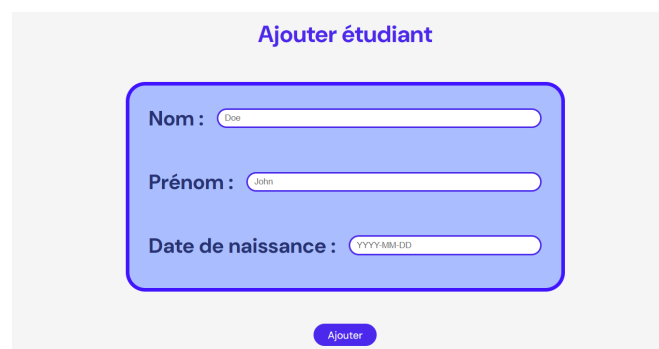


Page étudiants

Recherche Filtre Ajouter Étudiant

Nom	Prénom	Date de naissance	Contact	Promo	Action
Soussa	Anis	2003-07-05	soussaanis@cy-tech.fr	2027	<a href="#">✎</a> <a href="#">✖</a>
Paviet	Maxime	2002-12-01	pavietmaxi@cy-tech.fr	2027	<a href="#">✎</a> <a href="#">✖</a>
Guendouz	Sara	2003-02-12	guendouza@cy-tech.fr	2027	<a href="#">✎</a> <a href="#">✖</a>
Bouziane	Yasmina	2002-08-20	bouzianeya@cy-tech.fr	2027	<a href="#">✎</a> <a href="#">✖</a>
Zeboudj	Nawel	2003-07-24	zeboudjnaw@cy-tech.fr	2027	<a href="#">✎</a> <a href="#">✖</a>

Il peut alors faire une recherche à l'aide de la barre de recherche dynamique qui permet d'effectuer une recherche sur le nom, le prénom ou le contact. Il peut aussi modifier les informations d'un étudiant ou supprimer un étudiant grâce à la colonne "Action". Il peut appliquer un filtre qui permet de sélectionner les étudiants en fonction de leur "Promo". Enfin il peut créer un nouvel étudiant :



Ajouter étudiant

Nom :

Prénom :

Date de naissance :

S'il choisit la section "Cours". S'ouvre alors une page avec la liste de tous les cours :

**Page cours**

Recherche

Nom du cours

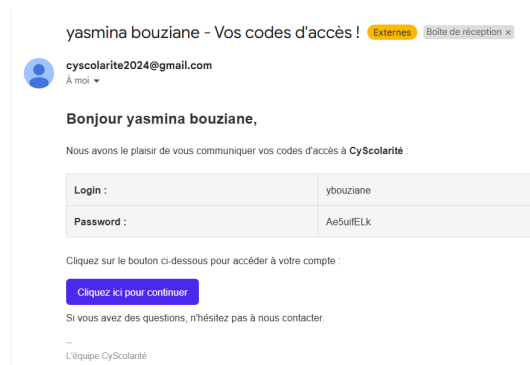
Ajouter cours

Cours	Professeur	Nombre d'élèves	Action
JEE	Mohamed Haddache	3	<a href="#">✎</a> <a href="#">🗑</a>
Cybersécurité	Ali Muhammad	3	<a href="#">✎</a> <a href="#">🗑</a>
Anglais	Aucun professeur	2	<a href="#">✎</a> <a href="#">🗑</a>
IA	Aucun professeur	3	<a href="#">✎</a> <a href="#">🗑</a>
Test et Verification	Aucun professeur	2	<a href="#">✎</a> <a href="#">🗑</a>
Architecture Réseau	Mohamed Haddache	2	<a href="#">✎</a> <a href="#">🗑</a>

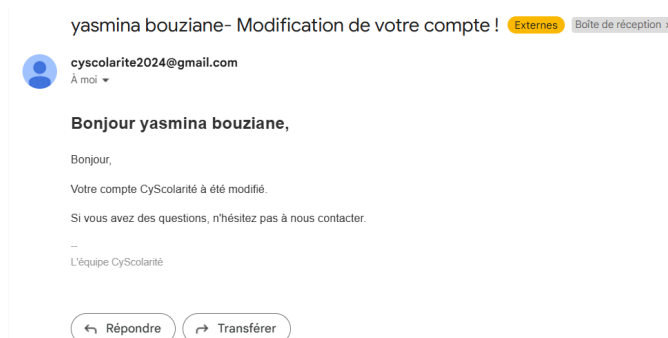
Une fois sur cette page, il peut faire une recherche à l'aide de la barre de recherche dynamique qui permet d'effectuer une recherche sur le nom du cours. Il peut aussi modifier les informations d'un cours ou supprimer un cours grâce à la colonne "Action". Enfin il peut créer un nouvel étudiant en renseignant le nom du nouveau cours dans la section "Nom du cours" puis en cliquant sur "Ajouter cours".:

### Envoie de mail:

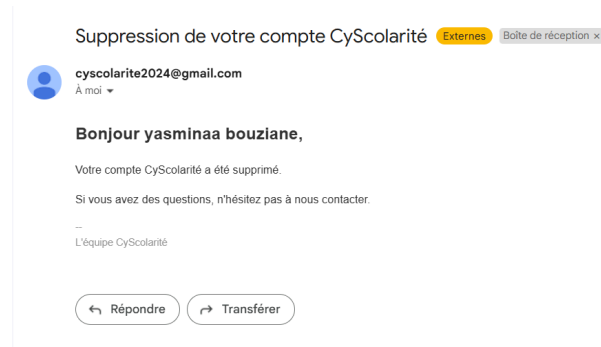
Un mail est envoyé à l'étudiant lorsque l'administrateur l'inscrit :



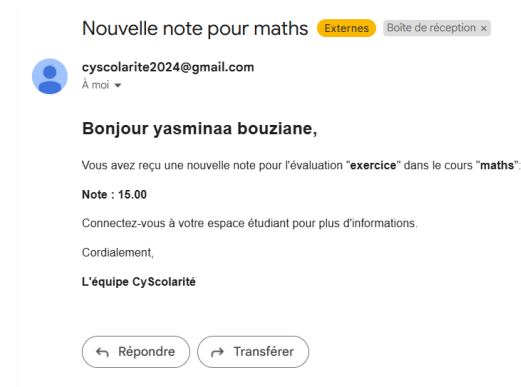
Un mail est envoyé à l'étudiant lorsque l'administrateur modifie une information le concernant :



Un mail est envoyé à l'étudiant lorsque l'administrateur supprime l'étudiant :



Un mail est envoyé lorsque l'enseignant attribue une nouvelle note à l'étudiant :



## Conception de l'application (SpringBoot) : (Branche springboot-main)

### Architecture Générale :

#### 1. Les principaux packages Java utilisés :

- **Package repositories** : Les repositories permettent de gérer les interactions avec la base de données. Ils facilitent les opérations CRUD (Créer, Lire, Mettre à jour, Supprimer) via des méthodes standard fournies par Spring Data JPA. Ils supportent aussi des requêtes personnalisées grâce à l'annotation @Query, simplifiant l'exécution de requêtes complexes
- 
- **Package controllers** : Contient les controllers responsables de gérer les interactions entre l'utilisateur et le système. Chaque controller correspond à une fonctionnalité spécifique comme par exemple `StudentPageAdminController`

ou `AddAssessmentTeacherController`. Ces classes contrôlent les requêtes HTTP et orchestrent la logique métier.

- **Package `models`** : Comprend les classes représentant les entités principales de l'application :
  - **`Student`** : représente un étudiant
  - **`Teacher`** : représente un enseignant,
  - **`Administrator`** : représente un administrateur,
  - **`Course`** : décrit un cours
  - **`Assessment`** : représente une évaluation pour un cours.
  - **`Grade`** : associer un étudiant à une évaluation avec une note
- **Package `services`**: Contient les classes responsables de la logique métier et de la communication avec la base de données :
  - **`StudentService`** : gère les opérations sur les étudiants (création, mise à jour, suppression).
  - **`TeacherService`** : gère les opérations sur les enseignants.
  - **`CourseService`** : gère les opérations liées aux cours.
  - **`AssessmentService`** : gère les évaluations.
  - **`GradeService`**: gère les notes.
  - **`AdministratorService`**: gère les génération et vérification des login/mots de passe.
  - **`EmailSenderService`** : gère le système de notifications par e-mail aux étudiants.

## 2. Diagramme de classe : (cf partie Jakarta)

## 3. Base de données :(cf partie Jakarta)

## Conclusion

### Bilan du projet :

Le développement de l'application de gestion de scolarité a permis de répondre aux principaux objectifs fixés, à savoir la simplification et l'optimisation de la gestion académique pour les administrateurs, enseignants et étudiants. Cette solution centralisée a été conçue pour automatiser les processus tels que l'inscription des

étudiants, l'attribution des cours, la saisie des notes et la consultation des résultats, tout en garantissant une expérience utilisateur fluide et sécurisée.

Grâce à une architecture robuste basée sur JEE et Hibernate, l'application assure une gestion efficace des données et une séparation claire des responsabilités via le modèle MVC.

### **Perspectives d'amélioration :**

Bien que l'application remplisse efficacement ses fonctions actuelles, des améliorations pourraient être apportées pour répondre encore mieux aux besoins des utilisateurs :

- **Personnalisation des notifications** : Permettre aux utilisateurs de configurer leurs préférences pour les notifications (type, fréquence, canal de communication).
- **Tableau de bord amélioré** : Ajouter des visualisations graphiques des performances des étudiants ou des statistiques liées aux cours.

### **Retour d'expérience :**

Le retour d'expérience sur ce projet a été globalement très positif. La collaboration au sein de l'équipe a permis une répartition efficace des tâches, assurant le respect des délais et la qualité du produit final. L'application a été testée dans divers scénarios pour garantir sa stabilité, son efficacité et son intuitivité.

La gestion rigoureuse des données, couplée à une interface utilisateur claire et à des fonctionnalités dynamiques, a contribué à fournir une solution complète et fiable. Ce projet a également permis de renforcer la compréhension et la maîtrise des technologies JEE, Hibernate et SpringBoot, tout en mettant en pratique des concepts clés tels que le modèle MVC et la gestion sécurisée des données.

En conclusion, cette application de gestion de scolarité représente une base solide sur laquelle il est possible de construire de nouvelles fonctionnalités. Elle s'inscrit dans une perspective d'évolution continue pour répondre aux exigences changeantes du domaine académique.