



# *PROJET CY-ENT*



BENZOUA Zinedine  
AWARE MAHAMAT Aware

MARTINS SOARES Flavio

KADDACHE Younes

EKOKA Elouan

CY-TECH

Projet JEE

ING2 Génie des Systèmes Informatiques 2024-2025

Monsieur HADDACHE Mohamed

# SOMMAIRE

<b>SOMMAIRE.....</b>	<b>2</b>
<b>PRÉSENTATION.....</b>	<b>3</b>
<b>RÉALISATION.....</b>	<b>4</b>
Développement.....	4
Choix techniques.....	4
Problèmes rencontrés.....	5
Solutions trouvées.....	5
Gestion des tâches.....	5
Spring Boot.....	6
<b>CONCEPTION.....</b>	<b>8</b>
Diagrammes UML.....	8
Diagramme de classe.....	8
<b>MANUEL D'UTILISATION (Application).....</b>	<b>9</b>
Point de vue admin.....	9
Point de vue étudiant.....	10
Point de vue professeur.....	11
Bonus.....	12
<b>BASE DE DONNÉES.....</b>	<b>14</b>
<b>BIBLIOGRAPHIE.....</b>	<b>15</b>

**CY ENT**

# PRÉSENTATION

Bonjour, nous sommes l'équipe de CY ENT, une équipe constituée de 5 étudiants en 2ème année de cycle ingénieur en génie des systèmes informatiques à CYTECH. Nous vous présentons dans ce communiqué le projet CY ENT. Qu'est-ce que CY ENT ?

Ce projet vise à concevoir une application web de gestion de scolarité adaptée aux besoins des étudiants, enseignants et administrateurs. L'objectif principal est de fournir une plateforme intuitive permettant une gestion efficace des informations académiques tout en respectant une architecture robuste et des technologies modernes. Les fonctionnalités incluent :

- **Gestion des étudiants** : Enregistrement, mise à jour, suppression, recherche, filtrage et consultation des détails des étudiants.
- **Gestion des enseignants** : Gestion des informations des enseignants et leur affectation aux cours.
- **Gestion des cours** : Création, modification, suppression et attribution des cours aux étudiants et enseignants.
- **Inscriptions et résultats** : Suivi des inscriptions, saisie des notes par les enseignants, génération de relevés de notes, consultation des résultats et calcul des moyennes.
- **Authentification et autorisation** : Sécurisation de l'accès aux fonctionnalités selon le rôle utilisateur (étudiant, enseignant, administrateur).

Le projet intègre des technologies comme **Java**, **Spring Boot**, **Hibernate**, **MySQL**, et respecte l'architecture **MVC** pour une meilleure modularité. Les données académiques seront stockées dans une base de données relationnelle avec une cohérence stricte, tandis que les interfaces graphiques seront développées en **JSP** pour une navigation fluide.

Des fonctionnalités avancées, telles que la génération de rapports de performance et les notifications par email, enrichiront l'expérience utilisateur. Ce projet mettra également l'accent sur la vérification des données avant tout enregistrement, garantissant leur validité et leur fiabilité.

L'objectif global est de créer une application dynamique et performante tout en assurant un haut niveau de qualité dans la conception, l'implémentation et la gestion des données.

# RÉALISATION

## Développement

### Choix techniques

Avant de commencer, il était nécessaire de choisir quel outil utiliser pour coder l'application et faire son visuel, créer les diagrammes et gérer l'organisation du projet.

Concernant la partie code, nous savions coder sous eclipse et IntelliJ. Ainsi il fallait choisir entre les deux.

Nous avons choisi d'utiliser IntelliJ IDEA comme environnement de développement intégré (IDE) pour ce projet. IntelliJ offre une gamme de fonctionnalités avancées qui facilitent le développement, telles que la complétion automatique du code, la navigation simplifiée, le débogage efficace et une intégration étroite avec les outils de gestion de versions comme Git.

Pour l'interface utilisateur de notre application, nous avons utilisé HTML, CSS, JavaScript.

Pour la gestion des données de notre application, nous avons opté pour une base de données relationnelle SQL. Cette décision a été motivée par la structure tabulaire des données que nous devons manipuler, ainsi que la flexibilité et la robustesse offertes par les systèmes de gestion de bases de données relationnelles.

Nous avons choisi d'utiliser une base de données SQL pour notre application en raison de plusieurs avantages clés. Tout d'abord, la structure tabulaire des données, avec des tables et des relations entre elles, correspond parfaitement au modèle relationnel des bases de données SQL. Cela facilite l'organisation et la gestion de nos données, qui incluent des informations sur les utilisateurs, les livres, et les emprunts. De plus, les bases de données SQL bénéficient d'une large adoption dans l'industrie, ce qui nous permet d'accéder à une vaste communauté de développeurs et à une abondance de ressources pour résoudre les problèmes et obtenir du support.

Enfin, les systèmes de gestion de bases de données relationnelles SQL sont connus pour leur robustesse et leur maturité, ayant été développés et affinés depuis des décennies. Cette maturité garantit une stabilité, une fiabilité et des performances éprouvées, assurant ainsi une base solide et fiable pour notre application de gestion de bibliothèque.

Nous avons choisi d'utiliser Wamp comme serveur pour notre base de données SQL. Wamp est une solution tout-en-un qui intègre Apache, MySQL et PHP, offrant ainsi un environnement de développement local facile à configurer et à utiliser. Cette solution nous permet de développer et de tester notre application localement avant de la déployer sur un serveur de production.

Il faut également savoir que nous utilisons le pilote JDBC MySQL Connector/J version 8.3.0 (mysql-connector-java-8.3.0.jar) pour permettre la connexion à notre base de données MySQL depuis notre application Java. Ce pilote fournit une interface standard JDBC pour interagir avec la base de données MySQL.

Le projet est développé en Java en raison de sa popularité, de sa portabilité et de sa robustesse. Java offre également une large gamme de bibliothèques et de frameworks qui facilitent le développement d'applications complexes.

Concernant la gestion du projet, nous avons utilisé le même outil que le projet de fin d'année d'ING 1: Jira. L'utilisation de Jira a grandement amélioré la gestion et l'organisation de notre projet CY ENT. Grâce à ses fonctionnalités de planification, de suivi, et de collaboration, nous avons pu maintenir une vue d'ensemble claire de notre progression, identifier rapidement les obstacles, et assurer une communication efficace au sein de l'équipe. Nous avons en plus de cela utilisé Discord, où nous avons créé un serveur qui nous a permis de nous rejoindre en vocal. Sans oublier GitHub qui nous a permis d'avoir une gestion du code efficace.

## Problèmes rencontrés

Au début du projet, l'organisation des sprints s'est révélée complexe. Nous avions du mal à estimer précisément la durée de chaque tâche, ce qui rendait difficile leur décomposition et leur attribution à différents membres de l'équipe. Cette incertitude a également créé des hésitations quant à l'ordre dans lequel aborder les tâches, ce qui a ralenti la progression initiale du projet.

## Solutions trouvées

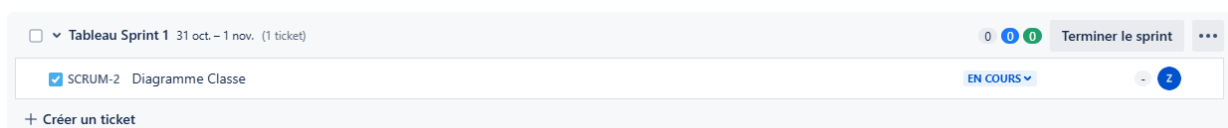
En avançant dans le projet, nous avons progressivement acquis une meilleure compréhension des tâches à réaliser et de leur complexité. Cela nous a permis de mieux identifier les tâches pouvant être décomposées en sous-tâches spécifiques et celles pouvant être réalisées en parallèle. Nous avons ajusté notre planification pour prioriser les tâches critiques au bon moment, évitant ainsi des retards ou des conflits dans la répartition des responsabilités. Cette amélioration a rendu nos sprints plus fluides et efficaces, renforçant la collaboration au sein de l'équipe.

## Gestion des tâches

Un jour après avoir choisi ensemble le sujet du projet, nous avons fait une liste des tâches à réaliser puis nous avons chacun pris ce qui nous plaisait. Voici ce qui en ressort:

- Aware :Partie SpringBoot, CRUD Enseignant, génération PDF.
- Zinedine : UML, JSP et CRUD Cours, Mailer.
- Flavio : Base de données, gestion d'erreurs, CRUD Utilisateur(Enseignant et étudiant).
- Younes : CRUD Inscription, JSP, CSS, HTML, vérifications et correction de bug.
- Elouan : Partie SpringBoot, Servlets déconnexion, jsp.

Voici comment nous avons décidé de répartir notre temps du début à la fin du projet à l'aide de Jira, pour plus de précision, nous avons défini quelle tâche allait être faite dans quel Sprint :



<input type="checkbox"/>	Tableau Sprint 2	2 nov. – 4 nov.	(7 tickets)	0	<span>0</span>	<span>0</span>	Terminer le sprint	...
<input checked="" type="checkbox"/>	SCRUM-4	Créer les tables dans la base de données en fonction du MCD. Configurer les mappings Hibernate pour chaque entité et tester la connexion.	EN COURS	-	A			
<input checked="" type="checkbox"/>	SCRUM-3	Créer les classes Java pour les entités (Étudiant, Enseignant, Cours, Inscription, Résultat) avec les relations appropriées.	EN COURS	-	A			
<input checked="" type="checkbox"/>	SCRUM-7	Mettre en place un système d'authentification et de gestion des rôles (étudiant, enseignant, administrateur).	EN COURS	-	J			
<input checked="" type="checkbox"/>	SCRUM-5	CRUD/servlet utilisateur	TERMINÉ(S)	-	FS			
<input checked="" type="checkbox"/>	SCRUM-6	Implémenter les contrôles de validation des données saisies avant leur enregistrement.	À FAIRE	-	Z			
<input checked="" type="checkbox"/>	SCRUM-19	CRUD/servlet Cours	À FAIRE	-	Z			
<input checked="" type="checkbox"/>	SCRUM-20	CRUD/servlet Inscription	À FAIRE	-	YK			
+ Créer un ticket								

<input type="checkbox"/>	Tableau Sprint 3	4 nov. – 8 nov.	(8 tickets)	0	<span>0</span>	<span>0</span>	Démarrer un sprint	...
<input type="checkbox"/>	SCRUM-9	Créer les interfaces JSP pour chaque rôle : étudiants, enseignants, administrateurs. Intégrer les formulaires pour la saisie et la mise à jour...	À FAIRE	-	J			...
<input checked="" type="checkbox"/>	SCRUM-10	Connecter les pages JSP avec les Servlets pour assurer les actions CRUD et les autres fonctionnalités (consultation, recherche, filtrage).	À FAIRE	-	YK			
<input checked="" type="checkbox"/>	SCRUM-11	Implémenter l'interface de saisie des notes et afficher les résultats.	À FAIRE	-	FS			
<input checked="" type="checkbox"/>	SCRUM-12	Ajouter les fonctionnalités de calcul des moyennes et génération de relevés de notes.	À FAIRE	-	A			
<input checked="" type="checkbox"/>	SCRUM-13	Intégrer une fonctionnalité de notifications par email pour les mises à jour des inscriptions et des notes.	À FAIRE	-	Z			
<input checked="" type="checkbox"/>	SCRUM-22	JSP Cours	À FAIRE	-	Z			
<input checked="" type="checkbox"/>	SCRUM-23	JSP Inscription	À FAIRE	-	YK			
<input checked="" type="checkbox"/>	SCRUM-24	JSP resultat	À FAIRE	-	J			
+ Créer un ticket								

<input type="checkbox"/>	Tableau Sprint 4	8 nov. – 15 nov.	(6 tickets)	0	<span>0</span>	<span>0</span>	Démarrer un sprint	...
<input checked="" type="checkbox"/>	SCRUM-25	JSP Utilisateur	À FAIRE	-	FS			
<input checked="" type="checkbox"/>	SCRUM-14	?Effectuer des tests complets sur les fonctionnalités principales (CRUD, saisie des notes, consultation des ...	À FAIRE	-	Z			
<input checked="" type="checkbox"/>	SCRUM-15	Migrer l'application vers Spring Boot en remplaçant les Servlets par des contrôleurs Spring.	À FAIRE	-	YK			
<input checked="" type="checkbox"/>	SCRUM-16	Intégrer Hibernate avec Spring Boot pour maintenir la persistance des données.	À FAIRE	-	A			
<input checked="" type="checkbox"/>	SCRUM-17	? Transférer la gestion des rôles et l'authentification à Spring Security pour plus de sécurité.	À FAIRE	-	FS			
<input checked="" type="checkbox"/>	SCRUM-18	? Publier la version finale de l'application sur GitHub et rendre le dépôt public avant la date de soumission.	À FAIRE	-	Z			
+ Créer un ticket								

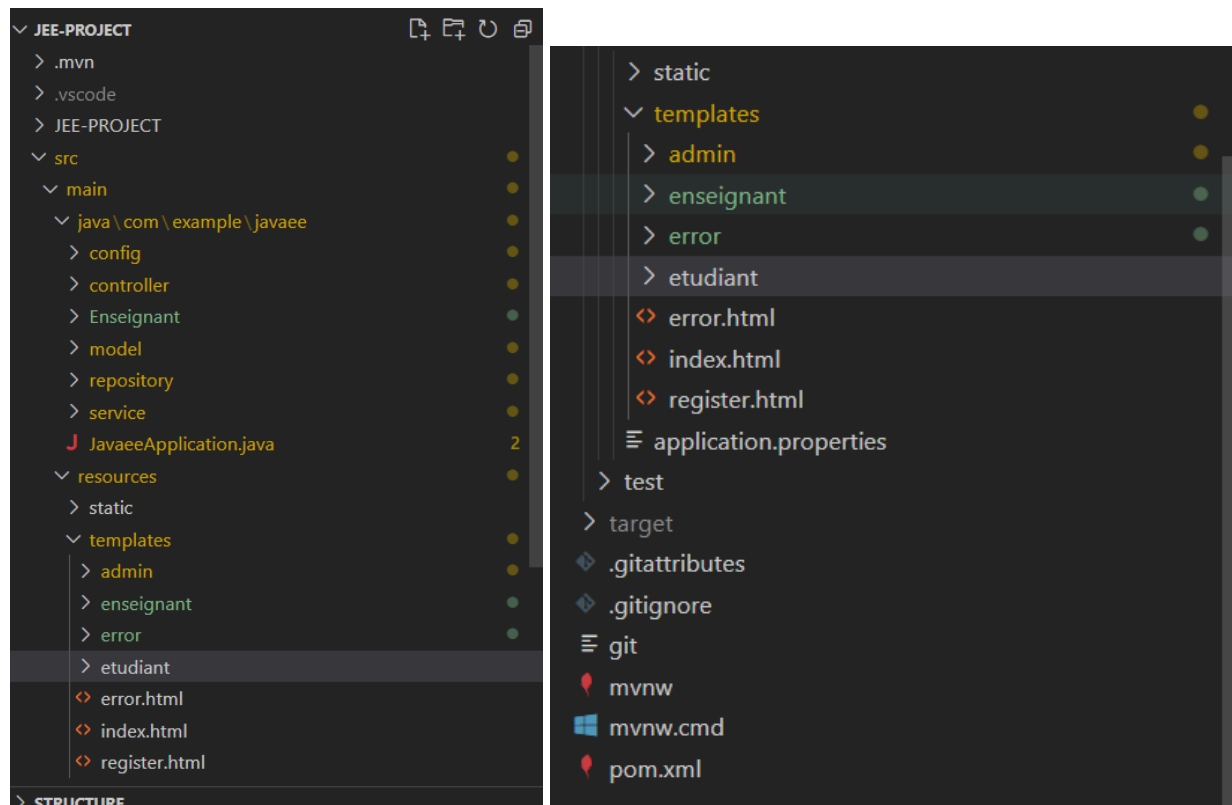
## Spring Boot

Il nous était imposé de convertir tout le projet en version Spring Boot, et cette démarche a été une étape clé dans l'amélioration de notre application. Le passage à Spring Boot nous a permis de simplifier énormément la configuration initiale du projet. Grâce à Spring Boot, nous avons pu réduire le nombre de fichiers de configuration nécessaires, tout en bénéficiant d'un démarrage automatique et rapide de l'application.

De plus, Spring Boot nous a offert une gestion intégrée des dépendances et une structure de projet plus claire et modulaire, ce qui facilite la maintenance et les évolutions futures. En résumé, cette conversion a non seulement rendu le projet plus robuste, mais elle a aussi amélioré la productivité de l'équipe en simplifiant plusieurs aspects techniques de la gestion du projet.

Nous avons ainsi modifié l'architecture du projet en supprimant les DAO et en passant l'architecture en modèle vue contrôleur.

Voici un exemple de l'arborescence du projet sous spring boot:



Faute de temps, nous n'avons pas fini de gérer les derniers détails sur la partie Spring Boot du projet. Il manque quelques modifications et vérifications de cohérence mais la partie Spring Boot est tout à fait fonctionnelle et utilisable dans l'extrême majorité du site. La version de base est simplement à un stade légèrement plus avancé et fonctionnel.

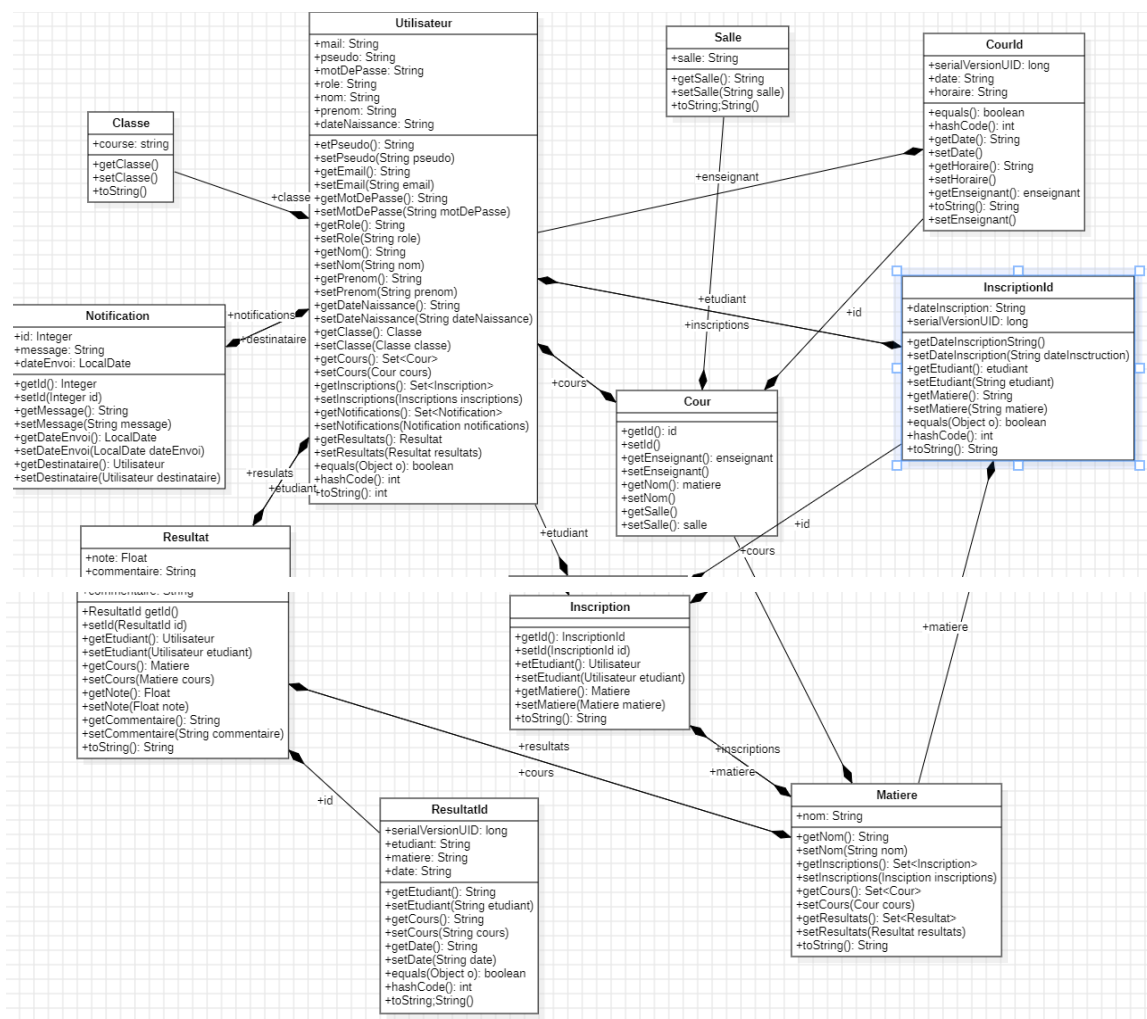
# CONCEPTION

## Diagrammes UML

Voici le diagramme UML que nous avons réalisé avant de commencer à coder le projet:

### Diagramme de classe

(insérer photo du diagramme)

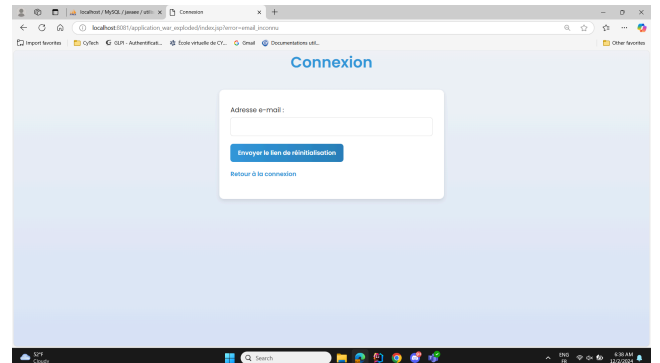
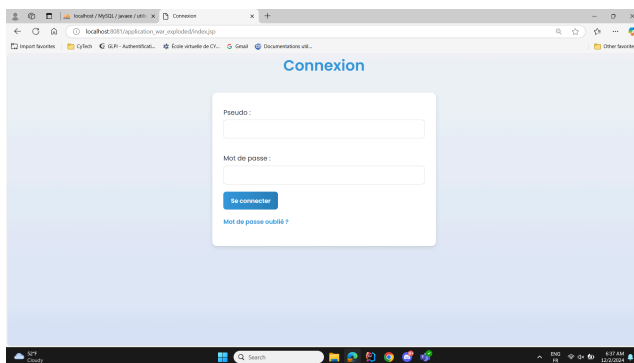




# MANUEL D'UTILISATION

## (Application)

Il faut d'abord savoir qu'à la page de connexion, si la personne a oublié son mot de passe, elle pourra simplement appuyer sur "mot de passe oublié" et insérer son adresse mail. Ainsi, elle recevra un nouveau mot de passe par mail grâce à notre magnifique mailer.



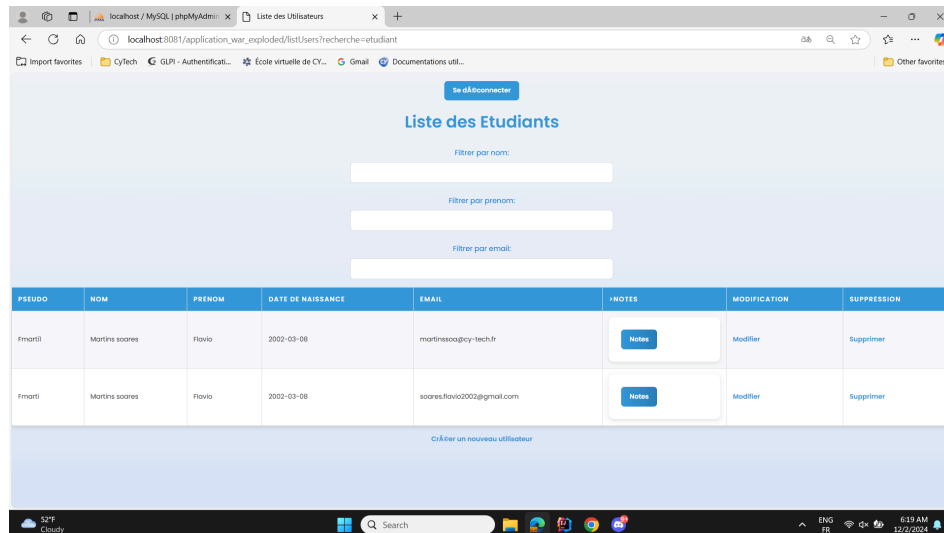
Pour se connecter en tant qu'admin, il faut mettre en pseudo admin et en mot de passe 6TkHmwVihU{@.

## Point de vue admin

Lorsqu'un administrateur se connecte, il dispose des fonctionnalités suivantes :

- Créer un étudiant ou un enseignant.
- Consulter la liste des enseignants et des étudiants.
- Ajouter une inscription ou un résultat, et consulter les listes correspondantes.
- Créer un cours et afficher la liste des cours.

De plus, l'administrateur peut modifier ou supprimer des informations depuis les différentes listes. Par exemple, dans la liste des étudiants, il est possible d'éditer ou de supprimer les données associées à un étudiant.



Dans chacune des pages "Liste de...", un système de filtrage est intégré, ce qui simplifie considérablement les recherches pour l'administrateur, notamment dans des cas où un très grand nombre d'étudiants ou d'éléments est affiché.

Cependant, dans la liste des cours, si l'administrateur souhaite modifier un cours, il ne peut intervenir que sur la date, l'horaire et la salle attribuée.

Lors de la création ou de la modification d'un cours, le programme effectue des vérifications automatiques pour garantir :

- Que le professeur est disponible au jour et à l'heure choisis.
- Que la salle sélectionnée n'est pas déjà réservée.

Ces validations assurent une gestion cohérente et évitent les conflits d'emploi du temps. Nous avons également fait en sorte que chaque cours dure 2h.

## Point de vue étudiant

L'option principale accessible à un étudiant depuis son espace ENT est l'inscription à un cours. En effet, l'étudiant peut facilement s'inscrire aux cours qu'il souhaite depuis son espace personnel.

En plus de cette fonctionnalité, l'étudiant a la possibilité de :

- Consulter son emploi du temps.
- Accéder à ses notes.
- Réinitialiser son mot de passe.
- Générer un fichier pdf avec ses notes et ses moyennes.. Vous trouverez un exemple ci-dessous:

## Bulletin de notes

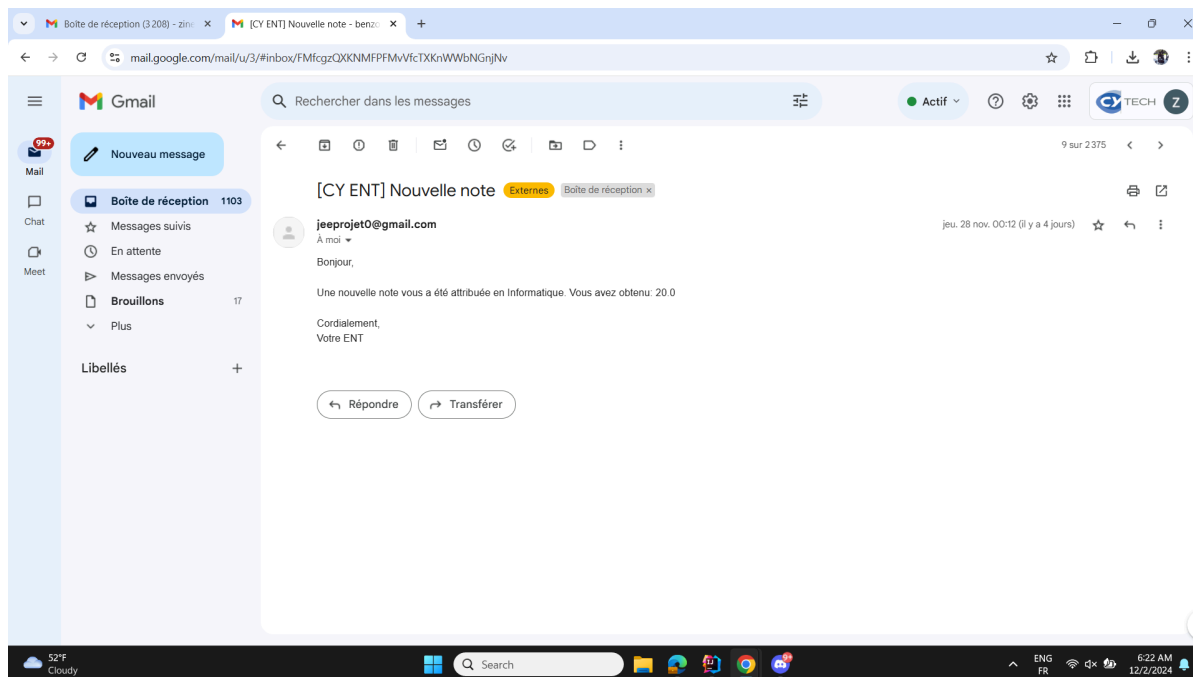
Nom : Kaddache Younes

Email : jeeprojet0@gmail.com

Matière	Moyenne du cours	Notes
Informatique	15,00	12.0, 13.0, 20.0
Moyenne Générale	15,00	

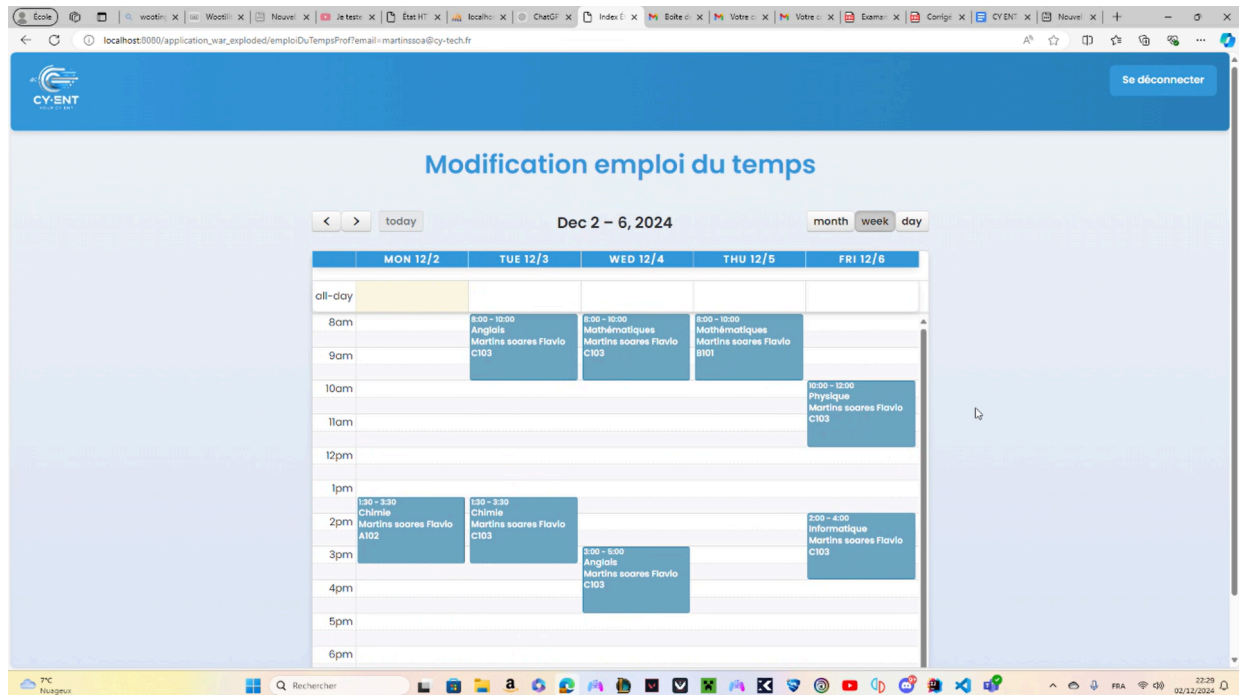
Lorsqu'une nouvelle note est ajoutée, qu'une note est modifiée, ou qu'un cours est annulé ou modifié, l'étudiant est immédiatement informé par e-mail.

Voici un exemple de mail automatique envoyé à l'étudiant :



## Point de vue professeur

Pour le professeur, il peut également voir son emploi du temps. Il peut aussi ajouter ou modifier une note aux étudiants inscrits sur ses différents cours. Le prof reçoit également un mail pour le tenir au courant quand un de ses cours est supprimé ou modifié. Voici l'emploi du temps que nous avons implémenté :



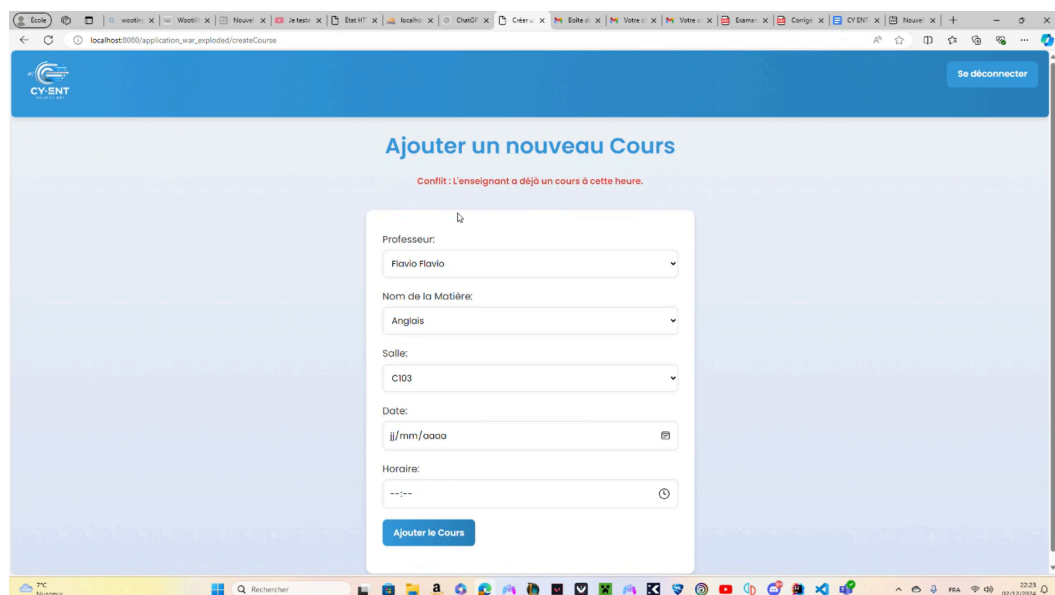
C'est aussi le type d'emploi du temps qu'ont les étudiants.

## Bonus

Plusieurs améliorations ont été apportées pour optimiser l'expérience utilisateur et la gestion des données :

- **Validation des formulaires et vérification** : Nous avons ajouté des vérifications afin de nous assurer que l'utilisateur insère un email valide dans le formulaire. Cette validation permet d'éviter les erreurs et garantit que les informations saisies sont correctes avant leur soumission.

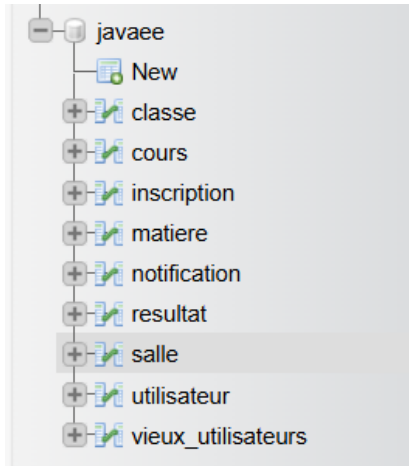
Vous trouverez ci-joint deux exemples de vérification : le premier lorsqu'une adresse email est invalide, et le second lorsqu'on tente d'attribuer un cours à un professeur alors qu'il a déjà un cours à cette heure-là. Dans les deux cas, les messages d'erreur correspondants s'affichent clairement pour informer l'utilisateur.



- **Base de données automatique** : Pour faciliter le démarrage du projet, nous avons configuré le système de manière à ce que la base de données se lance automatiquement au lancement du projet. Cela permet d'éviter toute manipulation manuelle et de simplifier le processus de démarrage.
- **Emploi du temps adaptatif** : Nous avons conçu un emploi du temps interactif et adaptatif. Il se met à jour en temps réel lorsque l'utilisateur supprime ou modifie un cours. Ainsi, l'emploi du temps reflète toujours les modifications effectuées sans nécessiter de rechargement manuel ou de mise à jour externe. L'admin peut cliquer sur les cours présents sur l'emploi du temps pour les modifier.
- **Hachage des mots de passe** : "Dans le souci de garantir la sécurité des données sensibles, nous avons mis en place un mécanisme de hachage des mots de passe. Ainsi, au lieu de stocker les mots de passe en clair, nous utilisons un algorithme de hachage sécurisé pour les transformer en une chaîne de caractères unique et irréversible. Cette approche permet de protéger les mots de passe des utilisateurs, même en cas d'accès non autorisé à la base de données.

# BASE DE DONNÉES

Voici l'architecture de notre base de données:



Comme évoqué précédemment, la spécificité avec notre base de données est que les mots de passe sont cryptés: cela renforce l'idée de sécurité et de robustesse que l'on souhaitait mettre en place au sein de notre application. Voici un exemple de ce que l'on voit sur la base de données concernant les mots de passes des utilisateurs:

La capture d'écran montre l'interface phpMyAdmin pour la base de données 'javaee'. La table 'utilisateur' est sélectionnée, et une requête SQL a été exécutée pour afficher les données. Les mots de passe sont effectivement cryptés, ce qui est visible dans la colonne 'motDePasse'.

	pseudo	motDePasse	role	nom	prenom	dateNaissance	email	classe
<input type="checkbox"/>	admin	\$2a105d8f2zyys5V7hpnrydowd8N8XLTG4HNgDuf8pOV...	admin	admin	admin	1111-11-11	admin-3@admin.com	NULL
<input type="checkbox"/>	enseignant	\$2a105d8f2zyys5V7hpnrydowd8N8XLTG4HNgDuf8pOV...	enseignant	Martins	scores	Flavio	2002-03-08	lucadachery@cy-tech.fr
<input type="checkbox"/>	etudiant	\$2a105d8f2zyys5V7hpnrydowd8N8XLTG4HNgDuf8pOV...	etudiant	Martins	scores	Flavio	2002-03-08	martinscores@cy-tech.fr
<input type="checkbox"/>	etudiant	\$2a105d8f2zyys5V7hpnrydowd8N8XLTG4HNgDuf8pOV...	etudiant	Martins	scores	Flavio	2002-03-08	scores.flavio2002@gmail.com

# BIBLIOGRAPHIE

- <https://github.com>
- <https://stackoverflow.com>
- <https://openai.com/blog/chatgpt>
- <https://developers.google.com/gmail/api/guides/sending?hl=fr>
- [Sending Email Using the Gmail API And Java - YouTube](#)
- [www.wampserver.com](http://www.wampserver.com)
- <https://www.mysql.com/products/connector%20/>
- <https://docs.oracle.com/en/java/>