



# Projet JEE : Application web de gestion de scolarité

UE – INFORMATIQUE 1

ING2 - GSI / 2024-2025

Professeur : HADDACHE Mohamed  
2024 - 2025

CLOVIS Betsaleel  
JULI Rithan  
LAZIZI Yassine  
LE Pascal  
MELAIMI Anis

# Sommaire

<b>Introduction .....</b>	<b>3</b>
<b>I / Analyse du projet .....</b>	<b>4</b>
1) Contexte.....	4
2) Sujet.....	4
<b>II / Partie technique .....</b>	<b>5</b>
1) Conception et structure globale du projet .....	5
2) Manuel Technique du Projet Gestion de scolarité.....	6
a) Technologies et Outils .....	6
b) Fonctionnement .....	7
<b>III / Partie travail et gestion de projet.....</b>	<b>14</b>
1) Organisation du travail dans le groupe .....	14
2) Migration vers Spring Boot .....	14
<b>Conclusion .....</b>	<b>15</b>

## Introduction

Le présent rapport met en lumière un projet de développement d'une application web de gestion de scolarité, réalisé dans le cadre du module **J2EE (ING2, GSI), Projet, 2024-2025**. L'objectif principal de ce projet est de concevoir une application complète et efficace permettant aux différents types d'utilisateurs (administrateur, enseignants, étudiants) de gérer et consulter les informations académiques de manière centralisée et intuitive.

Cette application vise à proposer une interface qui simplifie et automatise des tâches essentielles telles que la gestion des étudiants, des enseignants, la planification des cours, l'affichage de l'emploi du temps, l'inscription aux filières, la consultation des résultats, les notifications par messagerie, etc.

Le développement de cette application repose sur une combinaison de technologies telles que Java, HTML, CSS, JavaScript, ainsi que des frameworks comme Spring Boot et Hibernate. Ces outils permettent de concevoir une architecture robuste, tout en offrant une interface utilisateur ergonomique et accessible, avec une large gamme de fonctionnalités adaptées aux différents besoins.

Dans ce rapport, nous détaillerons les différentes étapes du processus de conception et de réalisation technique. Nous commencerons par analyser le sujet ainsi que les besoins des utilisateurs, avant de présenter le modèle conceptuel des données et le diagramme de classes. Par la suite, nous exposerons l'implémentation de l'application, en décrivant les principales fonctionnalités et les défis techniques rencontrés. De surcroît, nous aborderons la migration vers Spring Boot pour garantir une solution complète et fiable, ainsi que ses avantages pour une application JEE.

# I / Analyse du projet

## 1) Contexte

Ce projet a été réalisé dans le cadre de l'UE Informatique 1, au sein du module **JEE (ING2, GSI), Projet, 2024-2025** par notre équipe composée de CLOVIS Betsaleel, JULI Rithan, LAZIZI Yassine, LE Pascal et MELAIMI Anis. Nous avons travaillé de manière collaborative tout au long du projet pour concevoir une application web dynamique et fonctionnelle, répondant aux exigences techniques et pédagogiques établies.

L'objectif principal était d'appliquer les concepts théoriques et pratiques appris en cours, tels que la programmation Java, ainsi que la gestion des bases de données avec Hibernate. Ce projet a également permis de renforcer nos compétences en architecture logicielle en suivant le modèle MVC (Modèle-Vue-Contrôleur). De plus, ce travail nous a permis de mieux comprendre l'utilisation des frameworks comme Spring Boot, en particulier la simplification qu'il apporte dans la configuration du projet, la gestion des dépendances via Maven ou Gradle, et l'intégration rapide avec des technologies comme Hibernate pour le mapping objet-relationnel. Il nous a également permis de tirer parti des annotations Spring pour réduire le code standard et de profiter de ses outils intégrés, tel que Spring Data JPA pour faciliter les opérations sur la base de données.

Nous avons développé une application web de gestion de scolarité, qui répond aux besoins des étudiants, enseignants et administrateurs en centralisant et en automatisant la gestion des données académiques. Ce travail s'inscrit dans un environnement d'apprentissage simulant les conditions d'un projet professionnel réel, avec des délais précis, des objectifs clairs et une coordination d'équipe.

## 2) Sujet

Le projet vise à concevoir une application web de gestion académique complète. Elle doit permettre à chaque utilisateur d'accéder à certaines fonctionnalités en fonction de son rôle :

- **Étudiants** : Consulter leurs emplois du temps, leurs notes, télécharger leurs relevés de notes et être informés des mises à jour via une messagerie.
- **Enseignants** : de saisir les notes pour les matières qu'ils enseignent, de consulter leurs emplois du temps de cours et de gérer leurs interactions avec les étudiants via une messagerie.
- **Administrateurs** : de gérer les informations des étudiants, des enseignants, ainsi que les inscriptions aux filières et les emplois du temps.

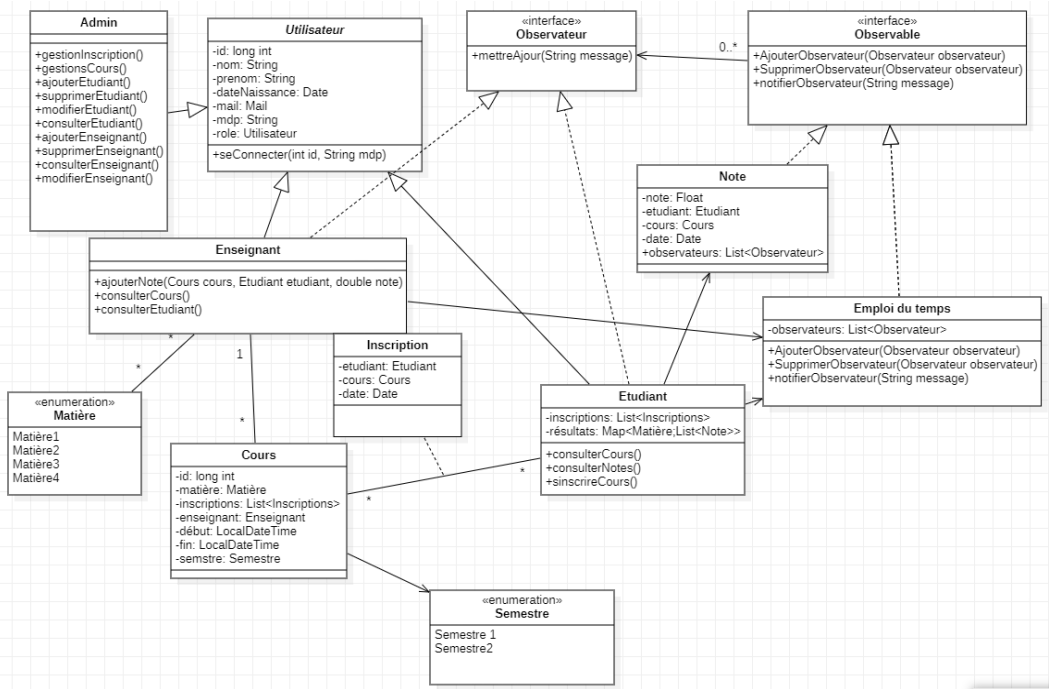
L'application repose sur les fonctionnalités suivantes, définies par le cahier des charges du projet :

- Gestion des utilisateurs (authentification et autorisation).
- Gestion des étudiants, enseignants et cours (CRUD).
- Inscription des étudiants aux cours.
- Saisie et gestion des résultats.
- Génération de rapports et de relevés de notes.
- Notifications via messagerie.

## II / Partie technique

### 1) Conception et structure globale du projet

- **Diagramme de classes**



- **Architecture**

Le projet suit une architecture **MVC** basée sur l'utilisation de modèles pour la gestion des données, de servlets comme contrôleurs, et de fichiers JSP pour les vues.

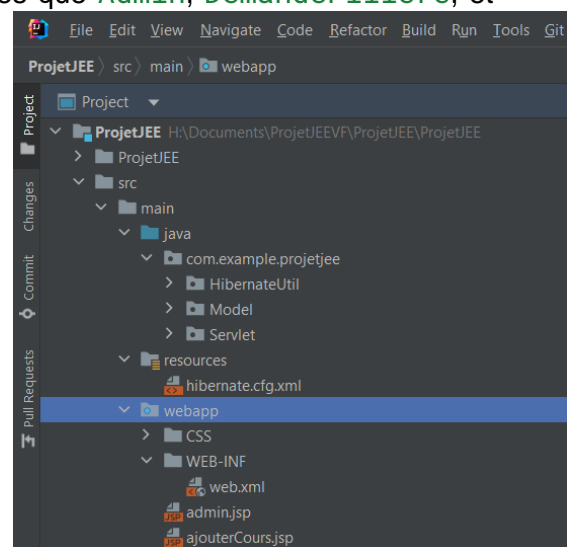
Le **package Modèle** regroupe les entités métiers, telles que **Admin**, **DemandeFiliere**, et **Enseignant**, mappées aux tables de la base de données.

Le package **WebApp(View)** présente les fichiers jsp qui seront affichés chez l'utilisateur tandis que les styles sont gérés via des fichiers CSS dédiés.

Les **servlets (Controller)**, situés dans le package correspondant, assurent la logique métier en traitant les requêtes des utilisateurs et en interagissant avec les entités Hibernate.

Le **package HibernateUtil** contient les configurations nécessaires pour gérer les connexions avec la base de données via le fichier **hibernate.cfg.xml**.

Cette structure garantit une séparation claire entre les responsabilités tout en exploitant la puissance d'Hibernate pour la persistance des données.



## 2) Manuel Technique du Projet C

### a) Technologies et outils

#### **Coder en Java sur IntelliJ IDEA :**

Nous avons choisi d'utiliser IntelliJ IDEA pour ce projet car cet IDE (Environnement de Développement Intégré) est particulièrement adapté au développement Java. Il offre de nombreuses fonctionnalités facilitant le travail, telles que des raccourcis, des suggestions de code intelligentes, une intégration avec Git et une gestion simplifiée des dépendances. De plus, c'est un outil avec lequel nous avons découvert Java, ce qui le rend familier et efficace pour notre équipe.

#### **Utilisation d'Hibernate :**

Hibernate a été sélectionné comme framework ORM (Object-Relational Mapping) pour simplifier la gestion de la base de données. Il permet de manipuler directement les objets Java tout en interagissant avec la base de données MySQL, sans avoir à écrire manuellement des requêtes SQL complexes. Hibernate lie directement les entités du package model aux tables de la base de données.

#### **Serveur Tomcat :**

Nous avons utilisé Apache Tomcat comme serveur d'applications. Il permet de déployer et d'exécuter des applications web basées sur des servlets et des JSP (JavaServer Pages). Tomcat est léger, rapide et bien intégré avec les outils de développement comme IntelliJ IDEA, ce qui en fait un choix naturel pour notre projet.

#### **Gestion de la base de données avec MySQL Workbench :**

Pour gérer notre base de données, nous avons opté pour MySQL Workbench. Il était déjà installé sur nos ordinateurs, ce qui a simplifié sa mise en œuvre. La base de données est connectée à l'application via Hibernate, et nous avons utilisé `hibernate.cfg.xml` pour configurer les paramètres de connexion.

#### **Framework JSP et CSS pour l'interface utilisateur :**

L'interface utilisateur a été développée en utilisant des JSP (JavaServer Pages) combinées avec des fichiers CSS. Les JSP permettent de créer des pages web dynamiques en intégrant des données issues de la base, tandis que CSS offre des styles visuels cohérents et modernes.

#### **Dépôt GitHub pour le contrôle des versions :**

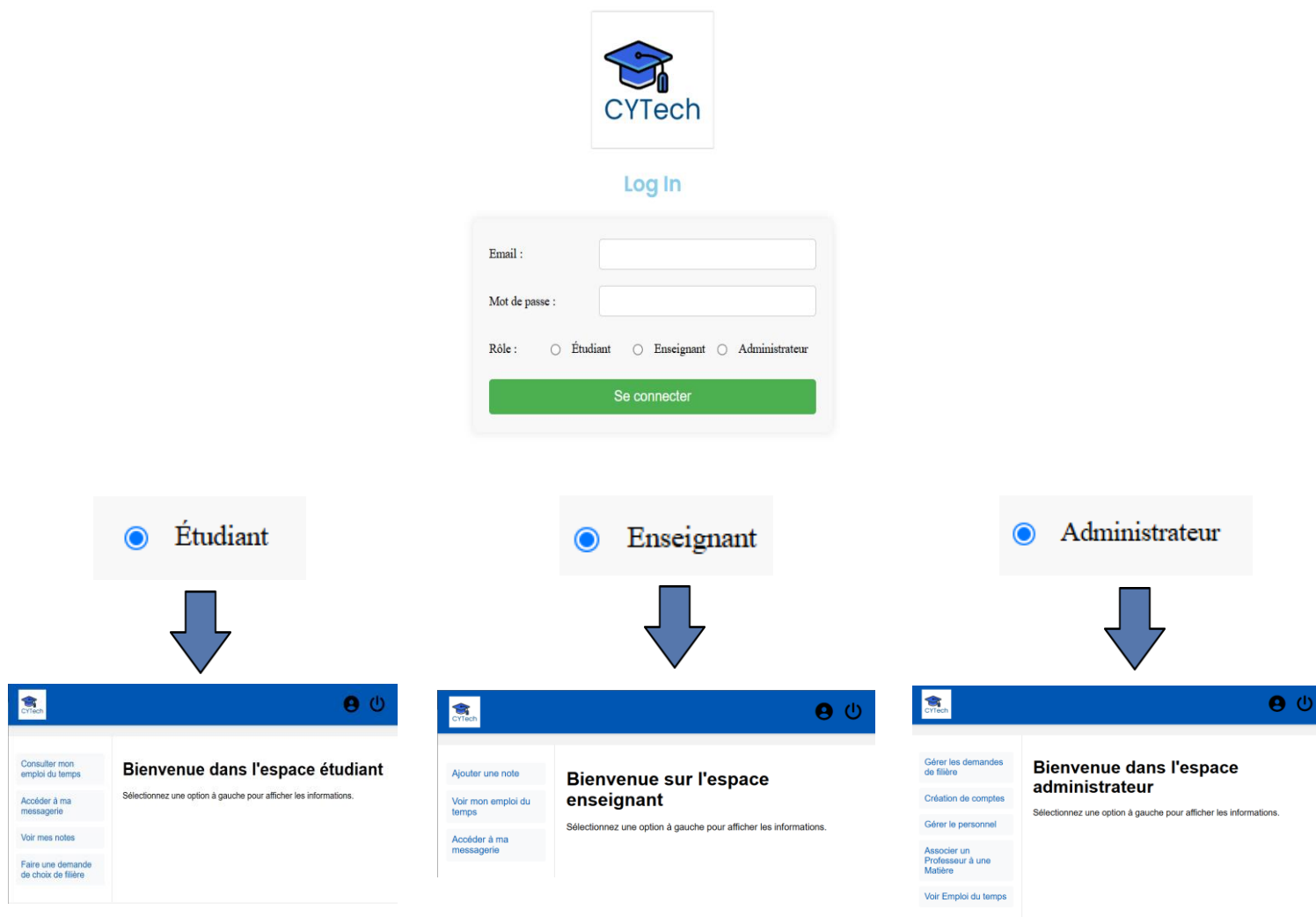
Un dépôt GitHub a été utilisé pour gérer les versions du projet et favoriser la collaboration au sein de l'équipe. Chaque membre a travaillé sur une branche dédiée, ce qui a permis d'intégrer leurs contributions dans une branche commune sans conflit. Git facilite également le suivi des modifications et la gestion des versions.

#### **Framework Spring Boot :**

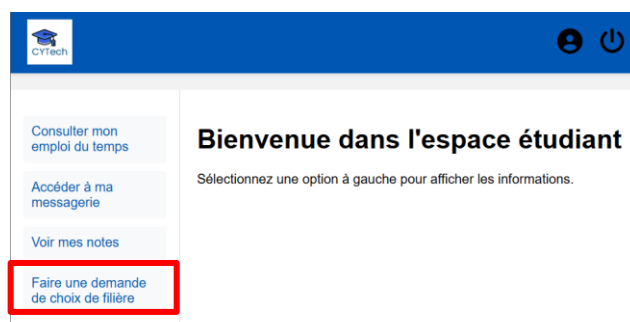
Le projet a migré vers Spring Boot afin de simplifier le développement, la gestion des dépendances et le déploiement. Spring Boot offre une structure modulaire et robuste qui nous a permis d'intégrer facilement des fonctionnalités avancées. Cette migration a également permis une meilleure organisation du projet, en séparant clairement les couches (Controller, Service, Repository).

## b) Fonctionnement

Voici la page d'authentification. Elle requiert l'email, le mot de passe et le rôle de l'utilisateur (étudiant, enseignant ou administrateur). Cette page est la première qui s'affiche lorsque l'on accède au site. Depuis celle-ci, il est possible d'accéder à trois espaces distincts : l'espace étudiant, l'espace enseignant et l'espace administrateur.



Voici la page de l'espace étudiant. Elle permet à l'étudiant de consulter son emploi du temps, d'accéder à sa messagerie, de visualiser ses notes et de faire une demande de choix de filière. Commençons par effectuer une demande de choix de filière auprès de l'administration pour l'élève [test.etudiant@gmail.com](mailto:test.etudiant@gmail.com).





## Choix de Filière

Choisissez une filière : Informatique

La demande de choix de filière de cet étudiant a été envoyée. Maintenant mettons-nous au point de vue de l'administrateur pour accepter cette demande de choix de filière:



Log In

Email : admin.test@gmail.com

Mot de passe : .....

Rôle : ☐ Étudiant ☐ Enseignant ☒ Administrateur



**Gérer les demandes de filière**  
Création de comptes  
Gérer le personnel  
Associer un Professeur à une Matière  
Voir Emploi du temps

**Bienvenue dans l'espace administrateur**  
Sélectionnez une option à gauche pour afficher les informations.



**Bienvenue dans l'espace administrateur**  
Sélectionnez une option à gauche pour afficher les informations.

**Liste des Demandes de Filière**

ID	EMAIL ETUDIANT	FILIÈRE	STATUT	DATE DE DEMANDE	COMMENTAIRE ADMIN	ACTIONS
1	etudiant.test@gmail.com	INFORMATIQUE	EN_ATTENTE	2024-11-30 13:25:00.0	En attente de traitement	<input type="button" value="Favorable à la demande"/> <input type="button" value="Ajouter/modifier le commentaire"/> <input type="button" value="Refuser"/> <input type="button" value="Supprimer"/> <input type="button" value="Accepter"/>

1



L'administrateur a bien reçu une demande de choix de filière de la part de cet étudiant. Il a la possibilité d'accepter, de refuser ou même de supprimer la demande. On peut également consulter les informations liées à la demande (ID, email de l'étudiant, filière demandée, statut de la demande, etc.) et ajouter un commentaire correspondant à cette demande. Nous avons également intégré une barre de recherche permettant de filtrer les demandes en fonction de la présence du mot-clé recherché dans l'un des attributs suivants : email de l'étudiant, filière, statut, commentaire ou date).

Dans notre cas, nous allons accepter la demande et ajouter un commentaire : *Favorable à la demande*. Nous pouvons constater que celle que nous avons acceptée a vu son statut mis à jour et qu'un commentaire y a été ajouté.

### Liste des Demandes de Filière

ID	EMAIL ÉTUDIANT	FILIÈRE	STATUT	DATE DE DEMANDE	COMMENTAIRE ADMIN	ACTIONS
1	etudiant.test@gmail.com	INFORMATIQUE	ACCEPTÉE	2024-11-30 13:25:00.0	Favorable à la demande	<input type="text" value="Commentaire admin..."/> <input type="button" value="Ajouter/Modifier le commentaire"/> <input type="button" value="Accepter"/> <input type="button" value="Refuser"/> <input type="button" value="Supprimer"/>

Toujours dans l'espace administrateur, nous avons la possibilité de créer des comptes de tout type :

Gérer les demandes de filière  
**Création de comptes**  
Gérer le personnel  
Associer un Professeur à une Matière  
Voir Emploi du temps

## Bienvenue dans l'espace administrateur

Sélectionnez une option à gauche pour afficher les informations.

### Création de Compte

Type de compte :  
Étudiant

Nom :

Prénom :

Email :  
etudiant.test@gmail.com

Date de Naissance :  
jj/mm/aaaa

Mot de Passe :  
\*\*\*\*\*

Créer le compte

Nous pouvons également gérer le personnel en accédant à une liste des étudiants ou des enseignants, avec la possibilité de supprimer ou de modifier leurs informations.

Gérer les demandes de filière  
Création de comptes  
**Gérer le personnel**  
Gérer les étudiants  
Associer un Professeur à une Matière  
Voir Emploi du temps

## Bienvenue dans l'espace administrateur

Sélectionnez une option à gauche pour afficher les informations.

### Liste des Enseignants

EMAIL	NOM	PRÉNOM	DATE DE NAISSANCE	ACTION
alaya.ines@gmail.com	Alaya	Ines	01/01/1990	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>
ali.muhammad@gmail.com	Ali	Muhammad	15/02/1985	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>
altair.yasin@gmail.com	Altair	Yasin	10/03/1987	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>
bouhathia.abderrahim@gmail.com	Bouhathia	Abderrahim	20/04/1983	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>

Il y a également une fonctionnalité qui permet d'associer un professeur à une matière, ainsi que de supprimer certaines associations si nécessaire.

**Bienvenue dans l'espace administrateur**  
Sélectionnez une option à gauche pour afficher les informations.

**Associer un Professeur à une Matière**

Professeur :  Matière :

**Associations existantes**

PROFESSEUR	MATIÈRE	ACTION
Alaya Ines	Optimisation déterministe	<input type="button" value="Supprimer"/>
Ali Muhammad	EDP	<input type="button" value="Supprimer"/>
Ali Muhammad	Cybersécurité Opérationnelle	<input type="button" value="Supprimer"/>
Alktar Yacin	Optimisation déterministe	<input type="button" value="Supprimer"/>

Enfin, la fonctionnalité "Voir emploi du temps" permet de consulter les emplois du temps associés aux filières Mathématiques et Informatique pour chaque semaine. Vous avez également la possibilité d'ajouter un cours à une semaine spécifique.

**Bienvenue dans l'espace administrateur**  
Sélectionnez une option à gauche pour afficher les informations.

**Emploi du Temps**

Choisir une filière :  Choisir une semaine :

Filière : null  
Semaine : null

Heures	Lundi	Mardi	Mercredi	Jeudi	Vendredi
Aucun emploi du temps disponible pour la sélection actuelle.					

On sélectionne la filière (par exemple, Informatique) et la semaine (comme la semaine 19), puis on clique sur "Afficher". Il est alors possible de visualiser l'emploi du temps dans lequel chaque cours peut être supprimé, déplacé ou un nouveau cours peut être ajouté.

**Emploi du Temps**

Choisir une filière :  Choisir une semaine :

Filière : Informatique  
Semaine : 19

Heures	Lundi	Mardi	Mercredi	Jeudi	Vendredi
08h-10h	IA: Applications (Sensouni) <input type="button" value="Supprimer"/> <input type="button" value="Déplacer"/>	Programmation C++ (Alktar) <input type="button" value="Supprimer"/> <input type="button" value="Déplacer"/>	Cybersécurité Opérationnelle (Ali) <input type="button" value="Supprimer"/> <input type="button" value="Déplacer"/>	Economie (Lefrioux) <input type="button" value="Supprimer"/> <input type="button" value="Déplacer"/>	Anglais (Majdoub) <input type="button" value="Supprimer"/> <input type="button" value="Déplacer"/>
10h-12h	Programmation fonctionnelle (Forest) <input type="button" value="Supprimer"/> <input type="button" value="Déplacer"/>	Programmation Système et Réseau (Haddache) <input type="button" value="Supprimer"/> <input type="button" value="Déplacer"/>	Aucun cours	Aucun cours	Economie (Lefrioux) <input type="button" value="Supprimer"/> <input type="button" value="Déplacer"/>
12h-14h	Pause	Pause	Pause	Pause	Pause
14h-16h	Architecture et Programmation Parallèle (Mercedal) <input type="button" value="Supprimer"/> <input type="button" value="Déplacer"/>	Aucun cours	Aucun cours	Aucun cours	Aucun cours
16h-18h	Aucun cours	Communication Interculturelle (Picod) <input type="button" value="Supprimer"/> <input type="button" value="Déplacer"/>	Aucun cours	IA: Applications (Sensouni) <input type="button" value="Supprimer"/> <input type="button" value="Déplacer"/>	Aucun cours

## Ajouter un Cours

Filière :  
Mathématiques

Semestre :  
1

Matière :  
Anglais

Professeur :  
Alaya Ines

Jour :  
Lundi

Heure :  
08h-10h

Semaine :  
Semaine 1

Ajouter

Retour à l'accueil

## Déplacer un Cours

Nouveau Jour :  
Lundi

Nouvelle Heure :  
08h-10h

Nouvelle Semaine :  
1

Déplacer

Retour à l'accueil

Voici la page de l'espace enseignant. Elle permet à l'enseignant de consulter son emploi du temps, d'accéder à sa messagerie, d'ajouter des notes aux et de faire une demande de choix de filière. Commençons par effectuer une demande de choix de filière auprès de l'administration pour l'élève [test.etudiant@gmail.com](mailto:test.etudiant@gmail.com).





Ajouter une note

Voir mon emploi du temps

Accéder à ma messagerie

## Bienvenue sur l'espace enseignant

Sélectionnez une option à gauche pour afficher les informations.

Accéder à ma messagerie

### Messagerie

Aucun message reçu.

#### Envoyer un nouveau message

Destinataire :  
Entrez l'email du destinataire

Sujet :  
Entrez le sujet du message

Contenu :  
Écrivez votre message ici...

Envoyer

Retour à l'accueil

Voir mon emploi du temps

### Emploi du Temps

Semaine : Semaine 8 [Changer de semaine](#)

Heures	Lundi	Mardi	Mercredi	Jeudi	Vendredi
08h-10h	Aucun cours	Aucun cours	Aucun cours	Aucun cours	Architecture réseau (Hadjilache)
10h-12h	Développement Distribué Java EE (Hadjilache)	Aucun cours	Architecture réseau (Hadjilache)	Aucun cours	Aucun cours
12h-14h	Pause	Pause	Pause	Pause	Pause
14h-16h	Aucun cours	Aucun cours	Aucun cours	Aucun cours	Aucun cours
16h-18h	Aucun cours	Aucun cours	Aucun cours	Aucun cours	Aucun cours

[Retour à l'accueil](#)

Ajouter une note



## Gestion des Notes

### Liste des Notes

NOM DE L'ÉLÈVE	MATIÈRE	NOTE
Etudiant	Développement Distribué Java EE	15.5

### Ajouter une Note

Email de l'élève :

etudiant.test@gmail.com

Matière :

Développement Distribué Java EE

Note :

15.5

Ajouter

Maintenant revenons à la page de l'espace étudiant. Elle permet à l'étudiant de consulter son emploi du temps, d'accéder à sa messagerie (messagerie qui contient des notification de nouvelle note), de voir ses notes et de faire une demande de choix de filière. Commençons par vérifier que cet étudiant a reçu une notification de nouvelle note pour la note qu'on vient de lui ajouter précédemment.



[Consulter mon emploi du temps](#)  
[Accéder à ma messagerie](#)  
[Voir mes notes](#)  
[Faire une demande de choix de filière](#)

## Bienvenue dans l'espace étudiant

Sélectionnez une option à gauche pour afficher les informations.



### Messagerie

Sujet : Nouvelle Note

De : haddache.mohamed@gmail.com

Reçu le : 2024-12-01T18:57:09

Vous avez reçu une nouvelle note : 15.50 en Développement Distribué Java EE.



[Consulter mon emploi du temps](#)  
[Accéder à ma messagerie](#)  
[Voir mes notes](#)  
[Faire une demande de choix de filière](#)

## Bienvenue dans l'espace étudiant

Sélectionnez une option à gauche pour afficher les informations.



Mes Notes	
Note (/20)	Date
Moyenne générale : 17.44 / 20	
Des résultats remarquables. Le travail et l'engagement sont exemplaires. Il faut continuer sur cette voie pour maintenir ce niveau d'excellence.	
Développement Distribué Java EE - Moyenne : 15.5 / 20	
15.5 / 20	10/11/2024
15.5 / 20	01/12/2024
Economie - Moyenne : 17.5 / 20	
17.5 / 20	15/11/2024
Optimisation déterministe - Moyenne : 18.2 / 20	
18.0 / 20	21/11/2024
20.0 / 20	21/11/2024
19.0 / 20	21/11/2024
19.0 / 20	21/11/2024
15.0 / 20	23/11/2024
<div>Télécharger le relevé de notes</div> <div>Retour à l'accueil</div>	



Relevé de Notes	
Nom : Etudiant	
Prénom : Test	
Date de naissance : 01/01/2000	
Moyenne générale : 17.44 / 20	
Des résultats remarquables. Le travail et l'engagement sont exemplaires. Il faut continuer sur cette voie pour maintenir ce niveau d'excellence.	
Economie (Moyenne : 17.5 / 20)	
Note (/20)	Date
17.5	2024-11-15
Développement Distribué Java EE (Moyenne : 15.5 / 20)	
Note (/20)	Date
15.5	2024-11-10
15.5	2024-12-01

<div> <div> <div>Consulter mon emploi du temps</div> <div>Accéder à ma messagerie</div> <div>Voir mes notes</div> <div>Faire une demande de choix de filière</div> </div> </div>	
<div> <div>Bienvenue dans l'espace étudiant</div> <div>Sélectionnez une option à gauche pour afficher les informations.</div> </div>	



## Emploi du Temps

Semaine : 

Semaine 1

Changer de semaine

Heures	Lundi	Mardi	Mercredi	Jedi	Vendredi
08h-10h	Statistiques (Bourhattas)	Design Patterns (Mercadal)	Cybersécurité Opérationnelle (Ali)	Anglais (Majdoub)	Design Patterns (Mercadal)
10h-12h	Développement Distribué Java EE (Haddache)	Test et Vérification (Forest)	Architecture réseau (Haddache)	Aucun cours	Communication Interculturelle (Picod)
12h-14h	Pause	Pause	Pause	Pause	Pause
14h-16h	Aucun cours	Aucun cours	Aucun cours	Statistiques (Bourhattas)	Aucun cours
16h-18h	IA: Théorie et Algorithmes (Senoussi)	Communication Interculturelle (Picod)	Aucun cours	Aucun cours	Aucun cours

Retour à l'accueil

### III. Partie travail et gestion de projet

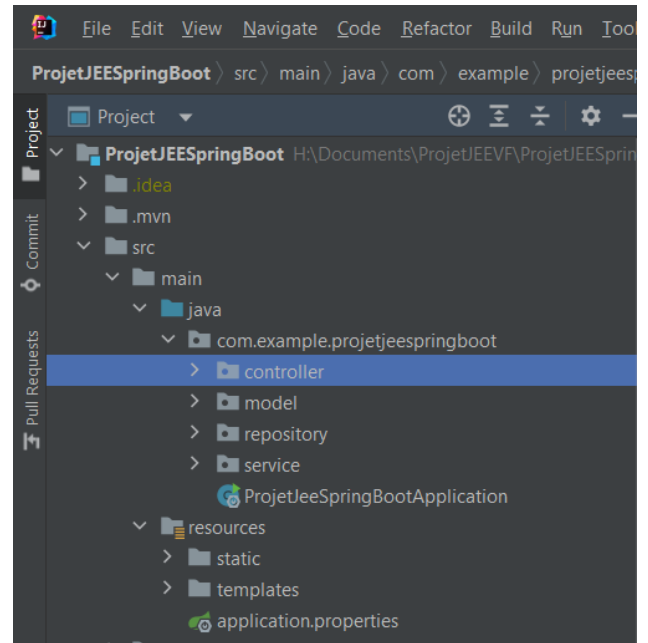
#### 1) Organisation du travail dans le groupe

Répartition des tâches du projet				
Fonctionnalité	Description	Tâches principales	Servlets associées	Membres
Gestion des Étudiants	Enregistrement, mise à jour, suppression, consultation des informations des étudiants	Création des entités, Servlets pour CRUD, JSP pour afficher/filtrer les données des étudiants	GererEtudiantsServlet ModifierEtudiantServlet SupprimerEtudiantServlet	Rithan
Gestion des Enseignants	Gestion des informations des enseignants, consultation et modification des détails	Développement des classes, Servlets pour CRUD, interfaces JSP pour gérer les enseignants	GererEnseignantsServlet ModifierEnseignantServlet SupprimerEnseignantServlet	Rithan Yassine
Gestion des Cours	Création, modification, suppression de cours, gestion des associations avec enseignants	Création des relations, Servlets pour CRUD, interfaces JSP pour afficher et gérer les cours	AjouterCoursServlet DeplacerCoursServlet SupprimerCoursServlet AssocierProfesseurMatiere SupprimerAssociationServlet	Pascal
Gestion des demandes de Filière	Inscription/désinscription des étudiants aux cours	Relation entre étudiants et filière, Servlets pour gérer les demandes de Filière	DemandeFiliereServlet GererDemandeFiliereServlet	Rithan
Gestion des Notes	Saisie par les enseignants, affichage des résultats par étudiant ou cours	Création des entités, Servlets pour gérer les notes, JSP pour saisir et consulter les résultats	AjouterNoteServlet NoteServlet ReleveNotesGenerator	Anis Betsaleel
Authentification et Rôles	Gestion des accès pour les administrateurs, enseignants et étudiants	Implémentation des rôles, Servlets pour authentification, pages JSP pour login et gestion de sessions	LoginServlet LogoutServlet CreerCompteServlet	Pascal
Emploi du Temps	Consultation des emplois du temps pour les étudiants et enseignants	Interfaces JSP et Servlets pour afficher les emplois du temps	AfficherEmploiDuTemps AfficherEDTEtuEnsServlet	Pascal Anis
Notifications et Messagerie	Envoi de messages et notifications pour les mises à jour importantes	Configuration du système de messagerie, développement des Servlets pour envoyer des notifications	MessageServlet	Anis Betsaleel
CSS et Js	Styles des pages et de l'application	Création des stylesheet CSS et fonctionnalités Javascript		Yassine Anis

#### 2) Migration du projet vers Spring Boot

Spring Boot est un **framework Java open-source** conçu pour simplifier et accélérer le développement d'applications web et microservices. Il repose sur le framework Spring et offre une configuration automatisée grâce à ses conventions par défaut, permettant aux développeurs de se concentrer sur la logique métier. Spring Boot intègre un serveur web embarqué (comme Tomcat ou Jetty) pour simplifier le déploiement des applications. De plus, il gère facilement les dépendances via des starters et offre un support intégré pour des fonctionnalités avancées comme l'authentification avec Spring Security ou la gestion de données avec Spring Data JPA.

Dans l'architecture d'un projet Spring Boot, chaque package a un rôle bien défini. Le **package model** définit les entités représentant les tables de la base de données. Le **package repository** gère l'accès aux données tandis que dans l'architecture JEE initiale, les **servlets** interagissent directement avec la base de données. Le **package service** contient la logique métier, centralisant les traitements complexes, tandis qu'en JEE, la logique métier serait souvent incluse dans les servlets. Le **package controller** dans Spring Boot joue un rôle similaire aux **servlets** dans un projet JEE sans avoir besoin de déclarer des **web.xml**. Enfin, le fichier `application.properties` centralise toutes les configurations de l'application, contrairement à JEE où les paramètres de configuration sont souvent dispersés dans des fichiers comme `web.xml` ou dans des classes de configuration spécifiques.



## IV) Conclusion

Pour conclure, nous avons réussi à développer une application web de gestion de scolarité qui est à la fois fonctionnelle, ergonomique et conforme aux exigences définies dans le cahier des charges. Cette application répond aux besoins des différents utilisateurs (étudiants, enseignants, administrateurs) grâce à ses nombreuses fonctionnalités, telles que la gestion des utilisateurs, des cours, des inscriptions, des notes, ainsi que la génération de relevés et l'envoi de notifications.

Le projet a impliqué le développement de plusieurs modules en Java, l'utilisation de **Hibernate** pour le mapping objet-relationnel, ainsi que la création et la gestion de la base de données avec **MySQL**. Nous avons également refait l'application en utilisant **Spring Boot** pour simplifier le développement backend. Les pages de vue ont permis de concevoir une interface utilisateur accessible et intuitive, répondant aux besoins des utilisateurs.

Au cours de ce projet, nous avons approfondi notre compréhension de la programmation orientée objet, la gestion de bases de données relationnelles, et l'architecture MVC (Modèle-Vue-Contrôleur). Nous avons également acquis une meilleure maîtrise des frameworks Spring Boot et Hibernate, tout en découvrant leur potentiel pour simplifier et optimiser le développement d'applications web. Spring Boot offre une approche plus modulaire et flexible que les projets JEE classiques, avec une architecture mieux structurée et une gestion plus fluide des composants.

Grâce à ce projet, nous avons non seulement mis en pratique nos compétences techniques en développement logiciel et en gestion de base de données, mais également renforcé nos capacités à collaborer en équipe. En utilisant des outils comme **Git** pour le versionnement du code et en respectant des délais précis, nous avons appris à organiser efficacement notre travail. Cette expérience nous a préparés à relever des défis similaires dans le cadre de futurs projets professionnels ou académiques.