

**Leo Nolland GSI1
Rayan Chaouche GSI3
Mathias Dufour GS3
Yssam Bairouki GS1**

J2EE (ING2, GSI)

2024-2025

Projet

Application web

Gestion de scolarité

Hadache Mohamed



Répartition des tâches :

- Leo Nolland : Springboot, Java et JSP
- Rayan Chaouche : Java, JSP et CSS
- Mathias Dufour : MySQL, CSS et JSP
- Yssam Bairouki : Java, MySQL et Springboot

Problèmes rencontrés

Beaucoup de problèmes lors de la partie 2 avec Hibernate, aucun de nous a réussi à le setup donc nous sommes passé sur IntelliJ, or seul un de nous 4 a réussi à activer sa license pour utiliser IntelliJ gratuitement, et seulement sur son PC fixe. Ce qui a énormément ralenti le travail, car pour la partie 2 on ne pouvait coder qu'avec une personne à la fois sur un seul PC.

A cause de ces problèmes, on s'est surtout concentrés sur faire fonctionner le projet sur la partie 2, et ajouter les détails dans la partie 3 (comme le CSS qui est donc disponible que dans la partie Springboot).

1. Introduction

Ce projet a pour objectif de développer une application web pour la gestion des informations académiques dans un établissement scolaire. Cette application doit permettre aux administrateurs, enseignants et étudiants de gérer les données académiques telles que les cours, les inscriptions, les résultats, et d'accéder à des fonctionnalités spécifiques selon leurs rôles.

1.1 Objectifs du projet

- Centraliser la gestion des données académiques.
- Fournir une interface utilisateur adaptée à chaque rôle (administrateur, enseignant, étudiant).
- Assurer la sécurité des données grâce à un système d'authentification et de gestion des rôles.
- Mettre en place une structure modulaire respectant l'architecture MVC.
- Permettre le suivi des performances des étudiants grâce à des fonctionnalités telles que la saisie des notes et le calcul des moyennes.

1.2 Technologies utilisées

- **Langages :**
 - **Back-end** : Java avec Spring Boot.
 - **Front-end** : JSP (Java Server Pages) avec HTML, CSS.
 - **Base de données** : SQL (MySQL).
- **Frameworks et bibliothèques :**

- Hibernate pour le mapping objet-relationnel (ORM).
 - Spring Boot pour la gestion des services et REST API.
 - Apache Tomcat comme serveur d'applications.
 - **Outils de développement :**
 - Maven pour la gestion des dépendances.
 - Eclipse IDE pour le développement.
-

2. Conception

La conception repose sur un **Modèle Conceptuel des Données (MCD)** qui définit les entités et leurs relations. Le schéma suivant, accompagné d'une analyse détaillée des classes Java, structure le projet.

2.1 Analyse des Entités et Relations

1. Entité : Cours

- **Attributs :**
 - `id` : Identifiant unique.
 - `dateCours` : Date et heure du cours.
 - `enseignant` : L'enseignant qui dispense ce cours.
 - `matiere` : La matière associée au cours.
- **Relations :**
 - Relation `ManyToOne` avec `Enseignant`.
 - Relation `ManyToOne` avec `Matiere`.
 - Relation `OneToMany` avec `Inscription` (plusieurs étudiants inscrits à un cours).

2. Entité : Enseignant

- **Attributs :**
 - `id` : Identifiant unique.
 - `nom` : Nom de l'enseignant.
 - `prenom` : Prénom de l'enseignant.
 - `contact` : Coordonnées de l'enseignant.
- **Relations :**
 - Relation `ManyToMany` avec `Matiere` (enseignant enseignant plusieurs matières).

3. Entité : Etudiant

- **Attributs :**
 - `id` : Identifiant unique.
 - `nom` : Nom de l'étudiant.
 - `prenom` : Prénom de l'étudiant.
 - `dateNaissance` : Date de naissance.

- `contact` : Coordonnées.
- **Relations :**
 - Relation `OneToMany` avec `Inscription` (étudiant inscrit à plusieurs cours).
 - Relation `OneToMany` avec `Note` (étudiant ayant plusieurs notes).

4. Entité : Inscription

- **Attributs :**
 - `id` : Clé composite (`etudiantId`, `coursId`).
 - `etudiant` : Référence vers l'étudiant inscrit.
 - `cours` : Référence vers le cours.
- **Relations :**
 - Relation `ManyToOne` avec `Etudiant`.
 - Relation `ManyToOne` avec `Cours`.

5. Entité : Matiere

- **Attributs :**
 - `id` : Identifiant unique.
 - `nom` : Nom de la matière.
- **Relations :**
 - Relation `ManyToMany` avec `Enseignant`.

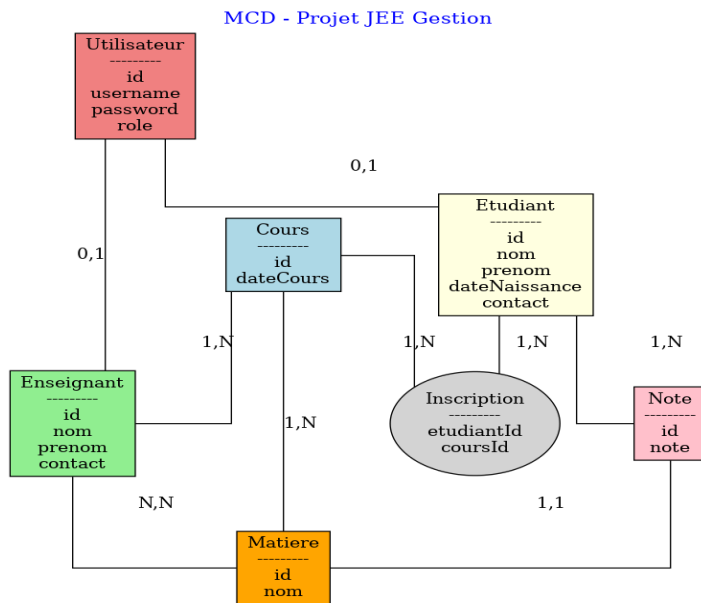
6. Entité : Note

- **Attributs :**
 - `id` : Identifiant unique.
 - `etudiant` : Référence vers l'étudiant ayant reçu la note.
 - `matiere` : Référence vers la matière associée.
 - `note` : Valeur de la note.
- **Relations :**
 - Relation `ManyToOne` avec `Etudiant`.
 - Relation `ManyToOne` avec `Matiere`.

7. Entité : Utilisateur

- **Attributs :**
 - `id` : Identifiant unique.
 - `username` : Nom d'utilisateur.
 - `password` : Mot de passe.
 - `role` : Rôle (`Administrateur`, `Enseignant`, `Etudiant`).
 - **Relations :**
 - Relation `ManyToOne` avec `Enseignant`.
 - Relation `ManyToOne` avec `Etudiant`.
-

2.2 Modèle Conceptuel des Données (MCD)



2.3 Schéma Relationnel

Basé sur le MCD et l'analyse des classes Java, le schéma relationnel est le suivant :

- Table `utilisateurs` : Stocke les données d'authentification.
- Table `etudiants` : Contient les informations personnelles des étudiants.
- Table `enseignants` : Contient les informations des enseignants.
- Table `cours` : Définit les cours disponibles.
- Table `matieres` : Liste les matières enseignées.
- Table `inscriptions` : Associe les étudiants aux cours.
- Table `notes` : Stocke les résultats académiques.

3. Implémentation

3.1 Architecture

L'application suit le modèle **MVC (Modèle-Vue-Contrôleur)** :

- **Modèle** : Les classes Java annotées avec JPA (Hibernate) représentant les entités.
- **Vue** : JSP (Java Server Pages) pour l'interface utilisateur.
- **Contrôleur** : Servlets et contrôleurs Spring Boot pour traiter les requêtes.

3.2 Fonctionnalités

- **Administrateur :**
 - Gestion des étudiants et enseignants.
 - Création, modification et suppression de cours.
 - Gestion des inscriptions.
 - **Enseignant :**
 - Attribution et modification de la date des cours.
 - Saisie des notes des étudiants.
 - Modifications des informations des étudiants
 - **Étudiant :**
 - Consultation des cours inscrits.
 - Consultation des résultats.
-
- **notes** sont attribuées aux étudiants pour une matière dans un cours spécifique.
-

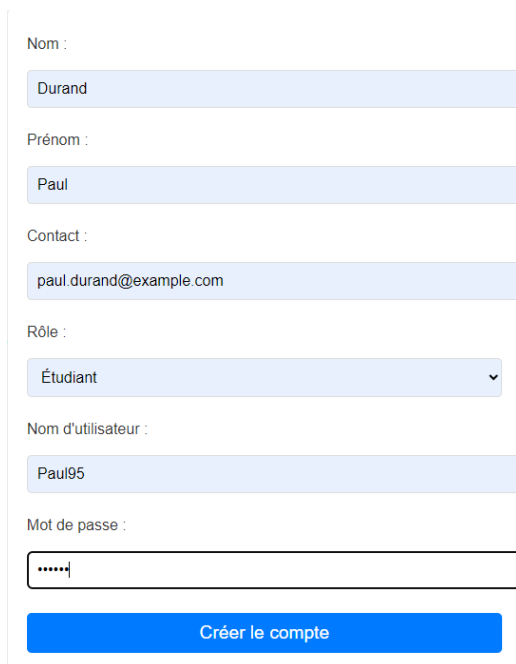
4. Résultats et Captures d'Écran

4.1 Captures d'écran de l'application

Les captures suivantes illustrent certaines des parties fonctionnelles de l'application :

4.1.1. Création de compte

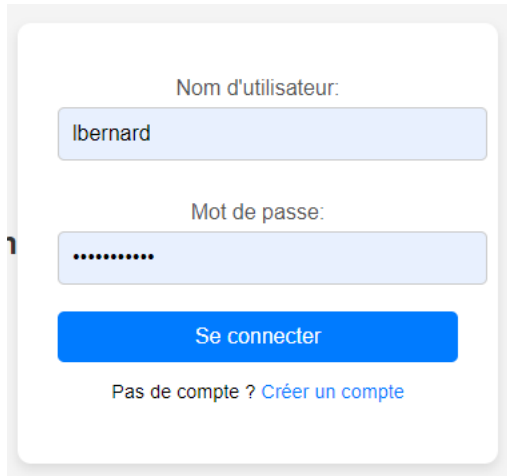
Permet de créer un compte étudiant ou enseignant seulement si il est déjà dans la base de données utilisateurs (pour éviter de créer des utilisateurs qui n'appartiennent pas à l'école).



The screenshot shows a web form for creating a new account. It contains several input fields with labels to their left: 'Nom :' with the value 'Durand', 'Prénom :' with 'Paul', 'Contact :' with 'paul.durand@example.com', 'Rôle :' with a dropdown menu showing 'Étudiant', 'Nom d'utilisateur :' with 'Paul95', and 'Mot de passe :'. The password field is masked with dots. At the bottom of the form is a blue button labeled 'Créer le compte'.

4.1.1. Écran de Connexion

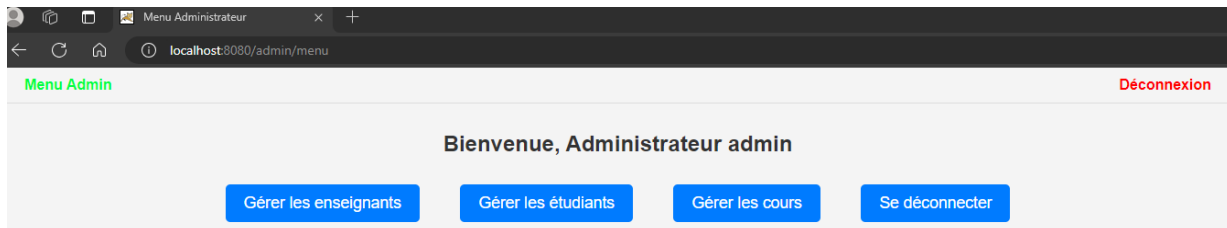
Cet écran permet aux utilisateurs (administrateurs, enseignants, étudiants) de se connecter via leurs identifiants. Une gestion des rôles est mise en place pour limiter l'accès selon le type d'utilisateur.



A login form with a light blue background. It contains two input fields: 'Nom d'utilisateur:' with the text 'lbernard' and 'Mot de passe:' with masked characters '.....'. Below the fields is a blue button labeled 'Se connecter'. At the bottom, there is a link 'Pas de compte ? Créer un compte'.

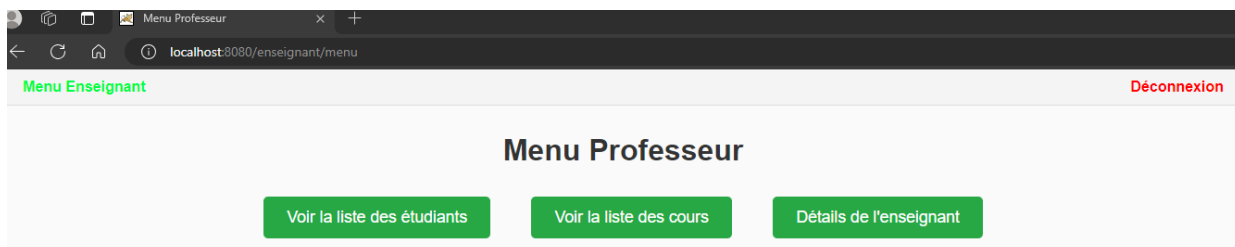
4.1.2. Interface Administrateur

L'administrateur peut gérer les étudiants, enseignants, cours et inscriptions. Les fonctionnalités incluent la création, modification, et suppression des enregistrements.



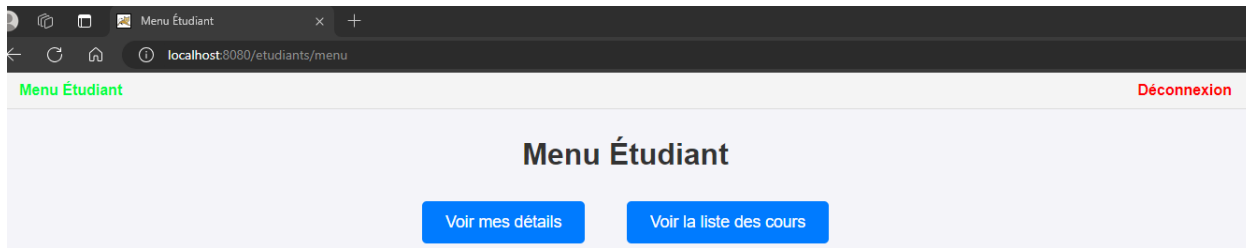
4.1.3. Interface Enseignant

L'enseignant peut consulter les cours qu'il enseigne, inscrire les étudiants et saisir leurs notes. Une liste des cours et des étudiants inscrits est affichée.



4.1.4. Interface Étudiant

L'étudiant peut consulter les cours auxquels il est inscrit, voir ses notes et accéder aux informations sur ses résultats académiques.



4.1.5. Gestion des cours

Un exemple de formulaire de création/modification d'un cours.

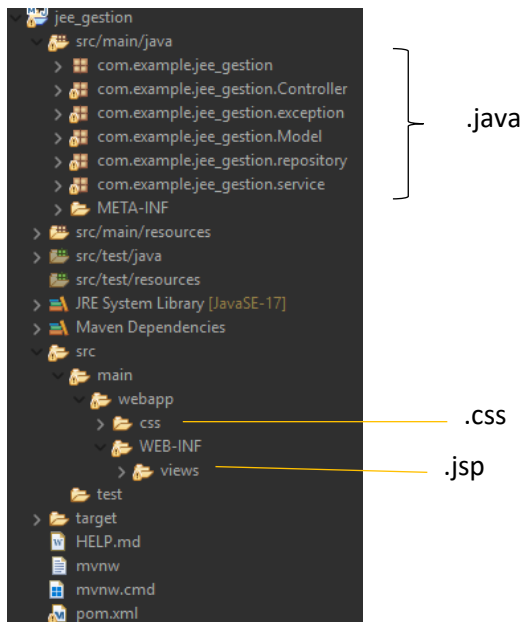
A screenshot of a form titled 'Ajouter un Cours'. It contains a 'Date et Heure :' label followed by a date and time input field with a placeholder 'jj/mm/aaaa --:--' and a calendar icon. Below this is a 'Matière :' label followed by a dropdown menu currently showing 'Mathématiques'. At the bottom of the form is a blue button labeled 'Ajouter'.

4.1.6. Gestion des notes

Les enseignants peuvent accéder à cette page pour saisir ou mettre à jour les notes des étudiants.

A screenshot of a form titled 'Ajouter une Note pour'. It features three input fields: 'Étudiant :' with a dropdown menu showing 'Martin Claire', 'Matière :' with a dropdown menu showing 'Mathématiques', and 'Note :' with a text input field containing the value '17'. A blue button labeled 'Ajouter' is positioned at the bottom of the form.

4.2. Captures d'écran du Package Explorer dans Eclipse



5. Types de Connexion et Rôles

5.1 Administrateur

- **Accès :**
 - Gestion complète des entités : étudiants, enseignants, cours, matières, inscriptions.
 - Création, modification et suppression des utilisateurs.
 - Accès aux rapports académiques (relevés de notes, statistiques).
- **Interface :**
 - Liste des entités avec options CRUD (Create, Read, Update, Delete).
 - Interface sécurisée avec gestion des autorisations.

5.2 Enseignant

- **Accès :**
 - Consultation des cours attribués.
 - Gestion des inscriptions pour les cours enseignés.
 - Saisie des notes pour les étudiants inscrits.
- **Interface :**
 - Accès limité aux données des cours qu'il enseigne.
 - Options pour ajouter ou modifier les notes des étudiants.

5.3 Étudiant

- **Accès :**
 - Consultation des cours inscrits.
 - Consultation des notes et résultats.

- Accès à ses informations personnelles.
- **Interface :**
 - Interface simple et intuitive affichant les résultats et les cours inscrits.
 - Formulaire de mise à jour des informations personnelles.

Annexes

Script SQL de la création de la BDD

```

1  -- Script de création de la base de données 'gestion_scolaire'
2
3  CREATE DATABASE IF NOT EXISTS gestion_scolaire;
4  USE gestion_scolaire;
5
6  -- Table 'enseignants'
7  CREATE TABLE `enseignants` (
8    `id` bigint NOT NULL AUTO_INCREMENT,
9    `nom` varchar(50) NOT NULL,
10   `prenom` varchar(50) NOT NULL,
11   `contact` varchar(100) DEFAULT NULL,
12   PRIMARY KEY (`id`)
13 ) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
14
15 -- Table enseignants_matieres
16 CREATE TABLE enseignants_matieres (
17   enseignant_id bigint NOT NULL,
18   matiere_id bigint NOT NULL,
19   PRIMARY KEY (enseignant_id, matiere_id),
20   KEY matiere_id (matiere_id),
21   CONSTRAINT enseignants_matieres_ibfk_1 FOREIGN KEY (enseignant_id) REFERENCES enseignants (`id`),
22   CONSTRAINT enseignants_matieres_ibfk_2 FOREIGN KEY (matiere_id) REFERENCES matieres (`id`)
23 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
24
25 -- Table 'matieres'
26 CREATE TABLE `matieres` (
27   `id` bigint NOT NULL AUTO_INCREMENT,
28   `nom` varchar(255) NOT NULL,
29   PRIMARY KEY (`id`)
30 ) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
31
32 -- Table 'cours'
33 CREATE TABLE `cours` (
34   `date_cours` datetime(6) DEFAULT NULL,
35   `enseignant_id` bigint DEFAULT NULL,
36   `id` bigint NOT NULL AUTO_INCREMENT,
37   `matiere_id` bigint DEFAULT NULL,
38   PRIMARY KEY (`id`),
39   KEY `FKf5ib2mIn0rs7m7x0dtssx1mk` (`matiere_id`),
40
41 -- Table 'inscriptions'
42 CREATE TABLE `inscriptions` (
43   `cours_id` bigint NOT NULL,
44   `etudiant_id` bigint NOT NULL,
45   PRIMARY KEY (`cours_id`, `etudiant_id`),
46   KEY `fk_inscriptions_etudiant` (`etudiant_id`),
47   CONSTRAINT `fk_inscriptions_cours` FOREIGN KEY (`cours_id`) REFERENCES `cours` (`id`),
48   CONSTRAINT `fk_inscriptions_etudiant` FOREIGN KEY (`etudiant_id`) REFERENCES `etudiants` (`id`),
49   CONSTRAINT `FKt3re4dbtj2w560ie98q0hhj1` FOREIGN KEY (`cours_id`) REFERENCES `cours` (`id`) ON DELETE CASCADE
50 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
51
52 -- Table 'resultats'
53 CREATE TABLE `resultats` (
54   `note` float DEFAULT NULL,
55   `etudiant_id` bigint DEFAULT NULL,
56   `id` bigint NOT NULL AUTO_INCREMENT,
57   `matiere_id` bigint DEFAULT NULL,
58   PRIMARY KEY (`id`),
59   KEY `FKjnhmxe4xedqwcs8iqbo32mhl` (`etudiant_id`),
60   KEY `FK5pobr70fbem9q23agg9piy4mr` (`matiere_id`),
61   CONSTRAINT `FK5pobr70fbem9q23agg9piy4mr` FOREIGN KEY (`matiere_id`) REFERENCES `matieres` (`id`),
62   CONSTRAINT `FKjnhmxe4xedqwcs8iqbo32mhl` FOREIGN KEY (`etudiant_id`) REFERENCES `etudiants` (`id`)
63 ) ENGINE=InnoDB AUTO_INCREMENT=26 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
64
65 -- Table 'utilisateurs'
66 CREATE TABLE `utilisateurs` (
67   `enseignant_id` bigint DEFAULT NULL,
68   `id` bigint NOT NULL AUTO_INCREMENT,
69   `password` varchar(255) DEFAULT NULL,
70   `username` varchar(255) DEFAULT NULL,
71   `role` enum('ADMINISTRATEUR','ENSEIGNANT','ETUDIANT') DEFAULT NULL,
72   `etudiant_id` bigint DEFAULT NULL,
73   PRIMARY KEY (`id`),
74   KEY `FK6252pe5k2mqpdnkjw7rdweauq` (`enseignant_id`),
75   KEY `FK_utilisateur_etudiant` (`etudiant_id`),
76   CONSTRAINT `FK6252pe5k2mqpdnkjw7rdweauq` FOREIGN KEY (`enseignant_id`) REFERENCES `enseignants` (`id`),
77   CONSTRAINT `FK_utilisateur_etudiant` FOREIGN KEY (`etudiant_id`) REFERENCES `etudiants` (`id`)
78 ) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

```

Lancement du projet avec Springboot

```

C:\Program Files\Java\jdk-21\bin\java.exe (29 nov 2024, 15:13:42) [pid: 19640]
:: Spring Boot ::      (v3.1.5)

2024-11-29T15:13:43.718401+00 INFO 19640 -- [see_gestion] restartedApplication c.w.jee.gestion.JeeDestinationApplication : Starting JeeDestinationApplication using Java 21.0.3 with PID 19640 (H:\Downloads\J...
2024-11-29T15:13:43.723401+00 INFO 19640 -- [see_gestion] restartedMain -w.c.jee.gestion.DefaultWebProcessor : No active profile set, falling back to default profile: "default".
2024-11-29T15:13:43.726401+00 INFO 19640 -- [see_gestion] restartedMain -w.c.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set '-spring.devtools.add-properties' to 'false' to disable.
2024-11-29T15:13:43.728401+00 INFO 19640 -- [see_gestion] restartedMain -w.org.springframework.boot.logging.LoggingWebWebProperties : For additional web related logging consider setting the 'logging.level.web' prop...
2024-11-29T15:13:44.684401+00 INFO 19640 -- [see_gestion] restartedMain -s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2024-11-29T15:13:45.291401+00 INFO 19640 -- [see_gestion] restartedMain -s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 182 ms. Found 7 JPA repository inter...
2024-11-29T15:13:45.691401+00 INFO 19640 -- [see_gestion] restartedMain -o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
2024-11-29T15:13:45.718401+00 INFO 19640 -- [see_gestion] restartedMain -o.apache.catalina.core.StandardService : Starting service [Tomcat]
2024-11-29T15:13:45.719401+00 INFO 19640 -- [see_gestion] restartedMain -o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.31]
2024-11-29T15:13:45.999401+00 INFO 19640 -- [see_gestion] restartedMain -org.apache.jasper.servlet.JspCompiler : At least one JAR was scanned for JIDs yet contained no URLs. Enable debug logg...
2024-11-29T15:13:46.008401+00 INFO 19640 -- [see_gestion] restartedMain -o.a.w.c.c.CContext[localhost]: Initializing Spring embedded WebApplicationContext
2024-11-29T15:13:46.358401+00 INFO 19640 -- [see_gestion] restartedMain -o.s.w.s.WebMultipartConfigurer : Initializing WebMultipartConfigurer in 221 ms
2024-11-29T15:13:46.246401+00 INFO 19640 -- [see_gestion] restartedMain -o.hibernate.jpa.internal.util.LogHelper : HHH000004: Processing PersistenceUnitInfo... (name: default)
2024-11-29T15:13:46.312401+00 INFO 19640 -- [see_gestion] restartedMain -org.hibernate.Version : HHH000042: Hibernate ORM core version 6.5.3.Final
2024-11-29T15:13:46.358401+00 INFO 19640 -- [see_gestion] restartedMain -o.s.c.i.internal.ResourceLoaderInitializer : HHH000025: Second level cache disabled
2024-11-29T15:13:46.691401+00 INFO 19640 -- [see_gestion] restartedMain -o.s.o.j.p.SpringPersistenceInfo : No LoadTimeWeaver setup; Ignoring JPA class transformer
2024-11-29T15:13:46.724401+00 INFO 19640 -- [see_gestion] restartedMain -com.zaxxer.hikari.HikariDataSource : HikariPool-1 Starting...
2024-11-29T15:13:47.166401+00 INFO 19640 -- [see_gestion] restartedMain -com.zaxxer.hikari.HikariPool : HikariPool-1 Added connection com.mysql.cj.jdbc.ConnectionImpl@1bfafac35
2024-11-29T15:13:47.167401+00 INFO 19640 -- [see_gestion] restartedMain -com.zaxxer.hikari.HikariDataSource : HikariPool-1 Start completed.
2024-11-29T15:13:47.248401+00 WARN 19640 -- [see_gestion] restartedMain -org.hibernate.orm.deprecation : HHH000082: MySQLDialect does not need to be specified explicitly using 'hibernate
2024-11-29T15:13:48.699401+00 INFO 19640 -- [see_gestion] restartedMain -o.n.e.t.j.p.i.PlatformInitiator : HHH000089: No JPA platform available (set 'hibernate.transaction.jpa.platform')
2024-11-29T15:13:48.673401+00 INFO 19640 -- [see_gestion] restartedMain -j.l.c.localContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2024-11-29T15:13:49.807401+00 INFO 19640 -- [see_gestion] restartedMain -s.d.r.j.query.QueryEnhancerFactory : Hibernate is in Classpath! If applicable, SQL parser will be used.
2024-11-29T15:13:50.178401+00 INFO 19640 -- [see_gestion] restartedMain -j.p.a.s.c.ConfigurationSpaceConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may m...
2024-11-29T15:13:50.676401+00 INFO 19640 -- [see_gestion] restartedMain -o.s.b.d.a.OptionalLiveLoadRunner : LiveLoadRunner is running on port 35729
2024-11-29T15:15:05.728401+00 INFO 19640 -- [see_gestion] restartedMain -o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context path '/'
2024-11-29T15:15:05.732401+00 INFO 19640 -- [see_gestion] restartedMain -s.d.r.c.RepositoryConfigurationDelegate : Started Spring Data repository scanning in 8183 ms

```