

# Projet JEE

## Application web de gestion de scolarité



Professeur référent: Mohamed HADDACHE  
Elèves : Alexandre RAZUMOWSKI  
Benjamin NEVEU  
Louis COURTOIS  
Giovanni MONTALTO  
Edgar FAUQUEMBERGUE  
ING 2 GSI  
2024-2025



# Sommaire

<b>Contexte et Objectifs du Projet</b>	<b>2</b>
<b>Fonctionnalités Clés</b>	<b>2</b>
<b>Étapes Réalisées dans la Version Servlets/JSP</b>	<b>3</b>
<b>Répartition des tâches :</b>	<b>3</b>
<b>Problèmes rencontrés :</b>	<b>4</b>

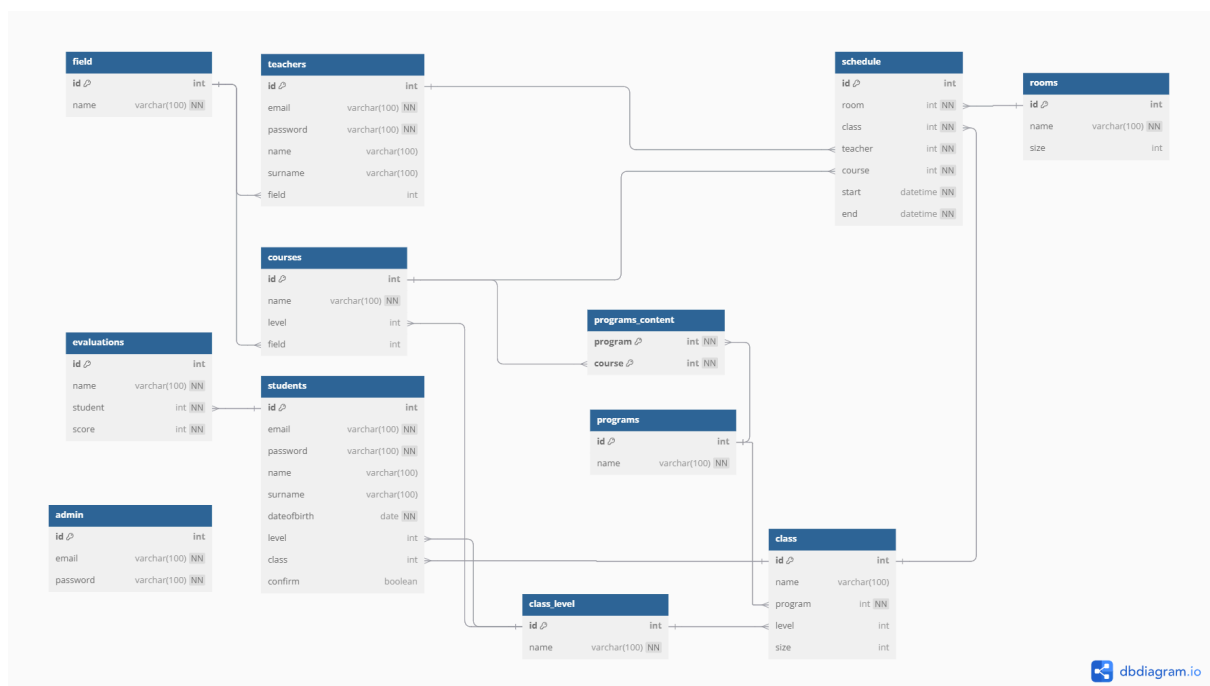
# Contexte et Objectifs du Projet

Le projet consiste à développer une application web pour la gestion de scolarité. L'application devait inclure des fonctionnalités pour la gestion des étudiants, enseignants, cours, inscriptions, et résultats. Elle est divisée en deux phases principales : implémentation initiale basée sur des Servlets et JSP, suivie d'une refonte en utilisant Spring Boot.

## Fonctionnalités Clés

1. Gestion des étudiants : CRUD sur les étudiants, recherche et filtrage.
2. Gestion des enseignants : Affectation des enseignants aux cours et CRUD.
3. Gestion des Cours : Création, modification et attribution des cours.
4. Inscriptions : Gestion des inscriptions aux cours.
5. Emploi du temps : Gestion et affichage des emplois du temps des étudiants et professeurs.
6. Gestion des Résultats : Saisie des notes, calcul des moyennes, et génération de relevés.
7. Sécurité : Authentification et autorisation basées sur les rôles (étudiant, enseignant, administrateur).

## MCD



# Étapes Réalisées dans la Version Servlets/JSP

## 1. Conception et Architecture

- a. Mise en place d'une architecture MVC avec :
  - i. **Modèle** : Entités Java (Hibernate utilisé pour le mapping).
  - ii. **Vue** : Pages JSP pour l'interface utilisateur.
  - iii. **Contrôleur** : Servlets pour gérer les requêtes HTTP.
- b. Création d'un MCD pour représenter les relations entre étudiants, enseignants, cours, inscriptions et résultats.

## 2. Base de Données

- a. Utilisation de MySQL pour la base de données.
- b. Modélisation des entités avec Hibernate (mapping des relations telles que les clés étrangères).

## 3. Développement Fonctionnel

- a. Implémentation des servlets pour chaque fonctionnalité principale.
- b. Validation des données dans les servlets avant insertion ou mise à jour.
- c. Création des JSP pour permettre une interaction fluide.

# Répartition des tâches

Nous nous sommes répartis les tâches de la façon suivante :

- Alexandre : Création du template et Validator, a travaillé principalement sur la partie de fonctionnement général du projet.

- Giovanni et Louis : Travail sur le fonctionnement de l'ajout des professeurs, élèves et emplois du temps, la gestion de la connexion, inscription et fonctionnement des rôles ci-dessus. Ils ont aussi travaillé sur la partie enregistrement des notes et affichage des notes en fonction des permissions.

- Edgar et Benjamin : Travail sur la partie matériel demandé, c'est-à-dire les classes, promotions, salles etc. Ils ont géré les liens entre les éléments comme le lien entre la promotion, la filière et les cours proposés.

## Problèmes rencontrés

Lors de notre projet, nous avons rencontré divers problèmes comme la création du Validator fonctionnel. Le but du Validator est de vérifier les champs entrés dans une zone d'écriture en précisant l'entrée attendue. Donc le type, requis ou non et avec une valeur max ce qui permet d'avoir un code passif qui vérifie de lui-même les entrées sans avoir besoin d'écrire beaucoup de ligne de code. Le Validator n'a pas été un problème en soit mais sa mise en place a pris beaucoup de temps ainsi qu'à quadriller le plus de cas possible et donc des possibles ajustements.

Un autre problème rencontré à été les liens entre les objets et les affichages en fonction d'une variable précise. Par exemple, un élève ne peut voir que ses notes et cela a posé quelques problèmes de logique dans le groupe mais rien de grave.

Nous avons aussi rencontré un problème avec l'envoi de mail. Nous pensons pour le coup que c'est un problème de configuration et non pas un problème de code.

Pour finir, notre plus gros problème à été le passage de Spring en Servelet. Nous avons fait le choix de commencer par faire le Spring en premier pour avoir une base solide mais lors de la conversion, nous avons eu beaucoup de mal à mettre en place la version Servelet de part le nombre de changement à faire et le fonctionnement très différents de Spring.

# Utilisation de Spring Boot

Objectifs :

- Simplifier le développement et la gestion du projet.
- Améliorer la modularité et la sécurité grâce aux fonctionnalités intégrées de Spring.

Actions Réalisées

1. **Migration de l'architecture vers Spring Boot :**
  - a. Utilisation de Spring MVC pour gérer les contrôleurs.
2. **Sécurité et Authentification :**
  - a. Configuration de Spring Security pour gérer les rôles et permissions.
  - b. Mise en place d'une gestion centralisée des sessions.
3. **Gestion de la Base de Données :**
  - a. Utilisation de Query SQL pour obtenir et modifier notre base de données en conséquence
4. **Nouvelles Fonctionnalités :**
  - a. Notification par email aux étudiants via l'API JavaMail.

## Comparaison des Approches

Aspect	Version Servlets/JSP	Version Spring Boot
Temps de Développement	Plus long (codage manuel)	Réduit (framework intégré)
Complexité	Elevée	Moyenne
Sécurité	Basique	Avancée (Spring Security)
Performances	Dépend de l'optimisation JSP	Optimisées
Extensibilité	Limitée	Haute

## Conclusion

Ce projet de gestion de scolarité nous a permis d'explorer deux approches : un développement initial en Spring Boot, rapide et structuré grâce à ses outils intégrés, suivi d'une migration vers Servlets/JSP. Cette transition a été complexe, nécessitant une réadaptation complète de l'architecture et une gestion plus manuelle des fonctionnalités. Malgré les défis, cette expérience a enrichi nos compétences techniques et notre compréhension des solutions web, tout en renforçant notre capacité à travailler en équipe face à des problématiques variées.