



Rapport de Projet

Système de Gestion Scolaire

Application Web - Spring Boot, JPA et Thymeleaf



Réalisé par : Aya Ahmyttou

Encadré par: Mr.EL HADDAD Mohamed

Lien github: https://github.com/ayyahmyttou/gestion_scolarite.git

Année Académique 2024-2025

Table des matières

Table des matières	2
Liste des figures.....	4
1. Introduction	5
2. Analyse des Besoins	5
2.1. Besoins Fonctionnels.....	5
2.1.1. Gestion des Filières	5
2.1.2. Gestion des Cours	5
2.1.3. Gestion des Élèves	5
2.1.4. Dossier Administratif	5
3. Conception.....	6
3.1. Diagramme de Cas d'Utilisation.....	6
3.2. Diagramme de Classes	6
3.2.1. Les 4 Entités Principales	7
3.2.2. Relations	7
4. Architecture Technique	8
4.1. Architecture en Couches (MVC)	8
4.1.1. Couche Présentation.....	8
4.1.2. Couche Contrôleur	8
4.1.3. Couche Service.....	8
4.1.4. Couche Persistance	8
4.1.5. Couche Données	8
4.1.6. Diagramme d'architecture	8
4.2. Dépendances Maven.....	9
4.2.1. spring-boot-starter-web	9
4.2.3. spring-boot-starter-data-jpa.....	9
4.2.4. spring-boot-starter-thymeleaf	9
4.2.5. mysql-connector.....	9
4.2.6. lombok	9
4.2.7. bootstrap via WebJars.....	9
5. Points Clés de l'Implémentation	10
5.1. Génération Automatique du Numéro d'Inscription.....	10
5.2. Gestion des Relations JPA	10
6. Réalisation - Démonstration de l'Application	11
6.1. Page d'Accueil.....	11
6.2. Gestion des Filières	11
6.2.1. Liste des Filières	11
6.2.2. Création d'une Filière	12
6.3. Gestion des Cours.....	13

6.3.1. Liste des Cours	13
6.3.2. Création d'un Cours	13
6.4. Gestion des Élèves.....	14
6.4.1. Liste des Élèves avec Recherche	14
6.4.2. Création d'un Élève	15
6.4.3. Profil Détailé d'un Élève.....	16
6.5. Messages et Notifications	17
7. Conclusion et perspectives	18

Liste des figures

Figure 1: diagramme de use case	6
Figure 2: diagramme de classe	6
Figure 3: Diagramme d'Architecture de l'application.....	9
Figure 4: page d'accueil	11
Figure 5: liste des filières	11
Figure 6: formulaire de création d'une filière.....	12
Figure 7: Message d'erreur si le code de filière existe déjà	12
Figure 8: Message de succès après création	12
Figure 9: Liste des cours.....	13
Figure 10: Formulaire de création d'un cours	13
Figure 11: Message d'erreur en cas de cours non attaché à une filière.....	14
Figure 12: Liste des élèves	14
Figure 13: Fonctionnalité de rechercher un élève.....	15
Figure 14: formulaire de création d'un élève.....	15
Figure 15: Profil détaillé d'un élève	16

1. Introduction

Ce rapport présente le développement d'une application Web de gestion scolaire complète réalisée avec Spring Boot. L'objectif est de créer un système permettant de gérer efficacement les élèves, filières, cours et dossiers administratifs.

Cette application permet de

- Centraliser la gestion des données scolaires
- Automatiser la création des dossiers administratifs
- Faciliter l'inscription des élèves aux cours
- Garantir l'intégrité et la cohérence des données

2. Analyse des Besoins

2.1. Besoins Fonctionnels

2.1.1. Gestion des Filières

- Créer, modifier, supprimer et consulter les filières
- Visualiser les élèves et cours associés
- Afficher des statistiques en temps réel

2.1.2. Gestion des Cours

- CRUD complet des cours
- Rattachement obligatoire à une filière
- Consultation de la liste des élèves inscrits

2.1.3. Gestion des Élèves

- CRUD complet avec recherche par nom/prénom
- Affectation à une filière
- Inscription aux cours de la filière
- Désinscription d'un cours

2.1.4. Dossier Administratif

Fonctionnalité critique :

- **Création automatique du dossier lors de l'ajout d'un élève**
- Génération automatique du numéro d'inscription : **FILIERE-ANNEE-ID**
Exemple : INFO-2025-12
- Enregistrement automatique de la date de création
- Association unique et permanente entre élève et dossier

3. Conception

3.1. Diagramme de Cas d'Utilisation

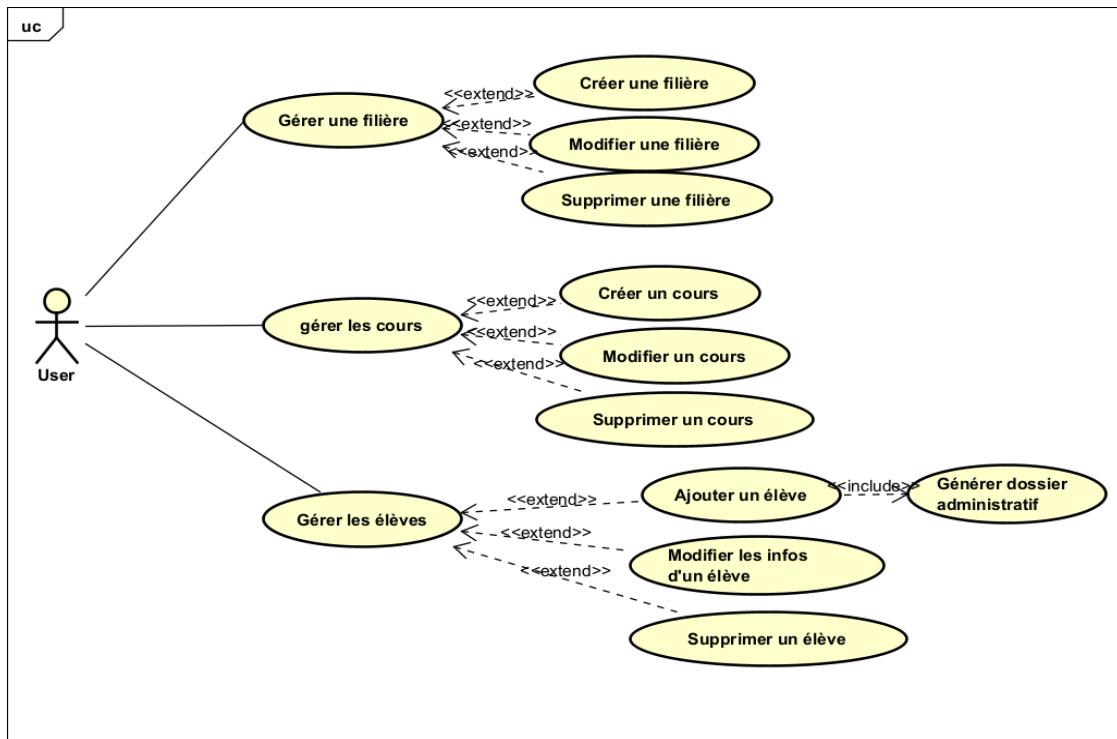


Figure 1: diagramme de use case

3.2. Diagramme de Classes

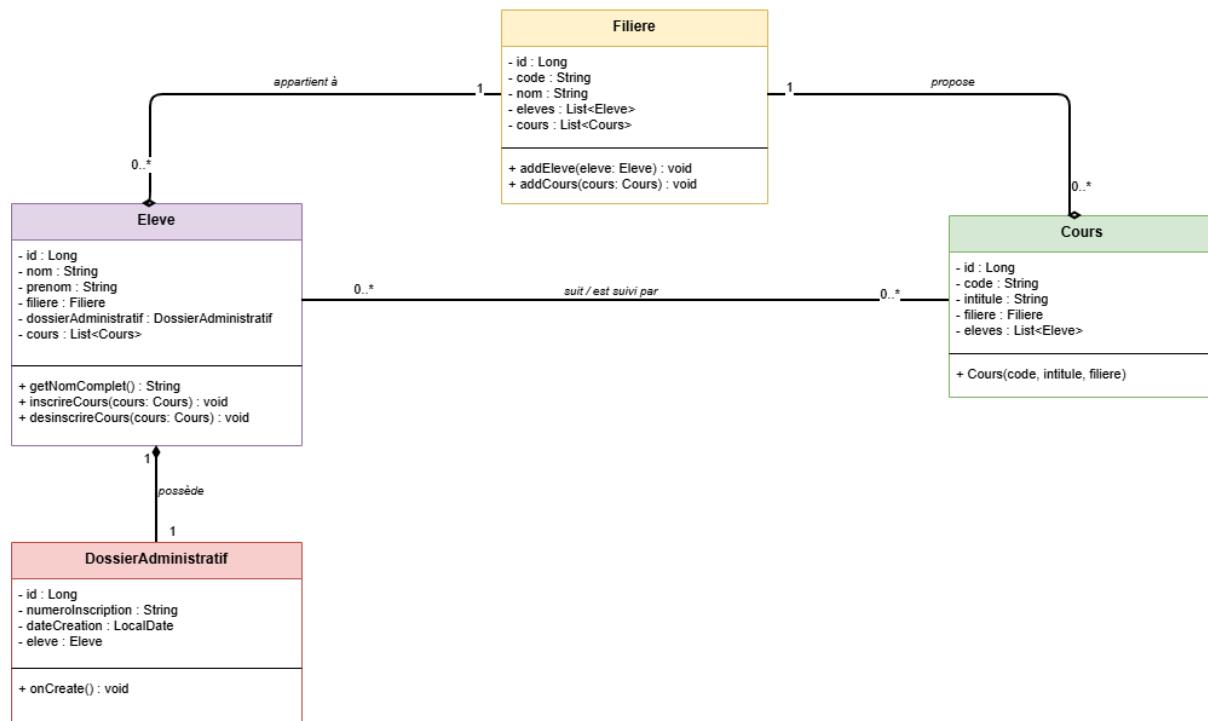


Figure 2: diagramme de classe

3.2.1. Les 4 Entités Principales

- **Eleve** : id, nom, prénom, filiere, dossierAdministratif, cours
- **Filiere** : id, code, nom, eleves, cours
- **Cours** : id, code, intitulé, filiere, eleves
- **DossierAdministratif** : id, numeroInscription, dateCreation, eleve

3.2.2. Relations

- **OneToOne** : Eleve ↔ DossierAdministratif (composition forte)
- **ManyToOne** : Eleve → Filiere, Cours → Filiere
- **OneToMany** : Filiere → Cours, Filiere → Eleves
- **ManyToMany** : Eleve ↔ Cours (table de jointure eleve_cours)

4. Architecture Technique

4.1. Architecture en Couches (MVC)

4.1.1. Couche Présentation

- **Technologies** : Thymeleaf + Bootstrap
- **Composants** : Templates HTML dynamiques, fragments réutilisables, CSS personnalisé

4.1.2. Couche Contrôleur

- **Technologies** : Spring MVC
- **Composants** : HomeController, EleveController, FiliereController, CoursController

4.1.3. Couche Service

- **Technologies** : Spring Service
- **Composants** : EleveService, FiliereService, CoursService, DossierAdministratifService

4.1.4. Couche Persistance

- **Technologies** : Spring Data JPA + Hibernate 6
- **Composants** : 4 Repositories avec méthodes de requête personnalisées

4.1.5. Couche Données

- **Base de données** : MySQL 8.0

4.1.6. Diagramme d'architecture

Le diagramme ci-dessous présente l'architecture globale de l'application en montrant toutes les couches du système et leurs interactions.

Cette architecture permet une séparation claire des responsabilités, réutilisabilité du code métier, une injection de dépendances avec Spring

Les différents couches sont :

- **Couche Entités (partie supérieure)** : Les 4 entités JPA du modèle de données avec l'ensemble de leurs attributs, méthodes et annotations JPA. On peut observer clairement les relations bidirectionnelles entre les entités.
- **Couche Repositories (partie gauche)** : Les 4 interfaces Spring Data JPA héritant de JpaRepository. Chaque repository expose des méthodes CRUD automatiques ainsi que des méthodes de requête personnalisées :
 - FiliereRepository : findByCode(), existsByCode()
 - CoursRepository : findByFiliere(), findByCode()
 - EleveRepository : findByNomContainingIgnoreCase(), findByFiliere()
 - DossierAdministratifRepository : findByNumerolInscription()
- **Couche Services (partie centrale)** : Les 4 services métier annotés qui implémentent la logique métier :
 - Gestion des transactions ACID
 - Validation des règles métier
 - Coordination entre repositories
- **Couche Controllers (partie droite)** : Les 4 contrôleurs Spring MVC annotés @Controller qui gèrent les requêtes HTTP :
 - Mapping des URLs (@GetMapping, @PostMapping)
 - Gestion du modèle pour les vues Thymeleaf
 - Validation des formulaires
 - Redirection et messages f

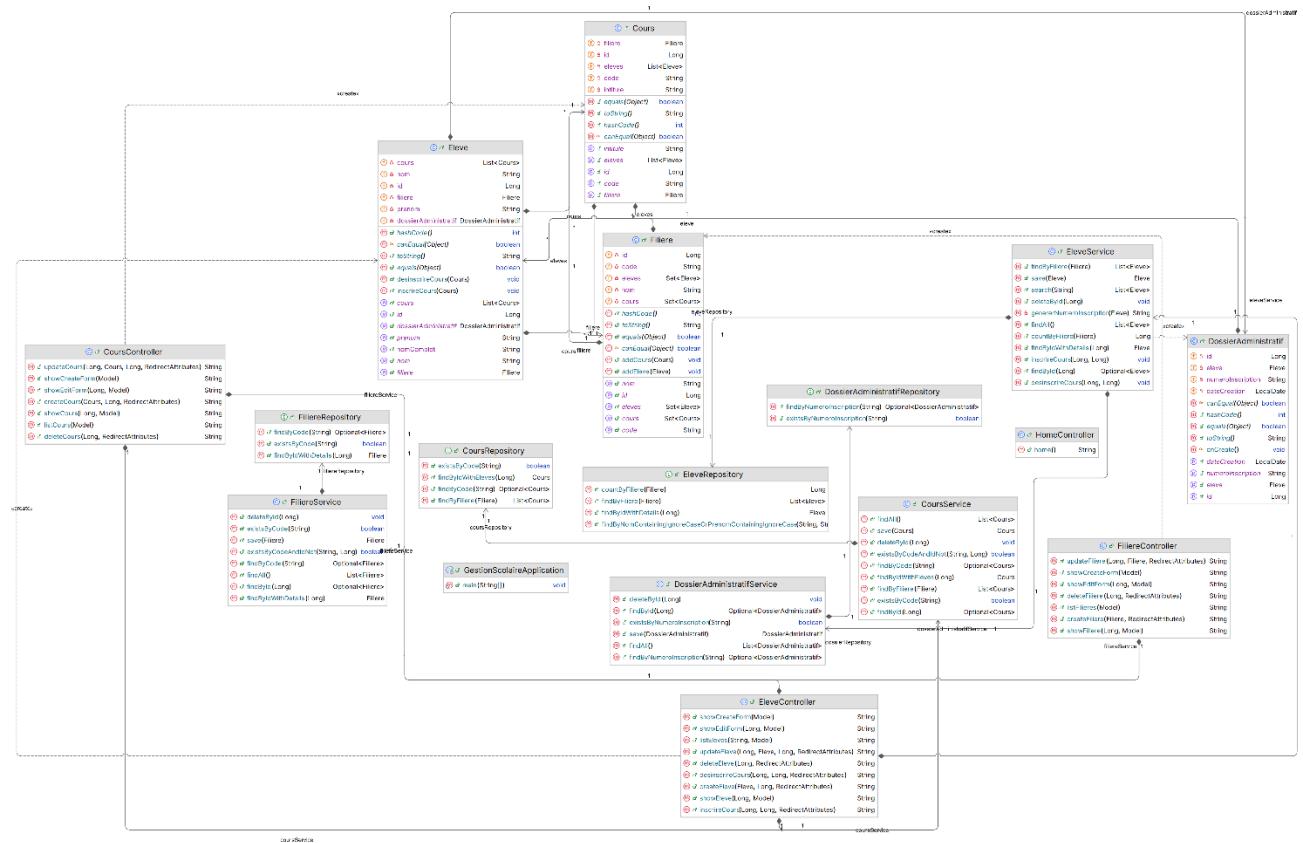


Figure 3: Diagramme d'Architecture de l'application

4.2. Dépendances Maven

4.2.1. spring-boot-starter-web

Permet de créer une application Web complète avec un serveur embarqué

4.2.3. spring-boot-starter-data-jpa

Gère la persistance des données et la communication avec la base de données. Sans JPA/Hibernate, il faudrait écrire tout le SQL manuellement et gérer les connexions à la main.

4.2.4. spring-boot-starter-thymeleaf

Moteur de templates pour générer des pages HTML dynamiques côté serveur.

4.2.5. mysql-connector

Driver JDBC pour connecter l'application Java à la base de données MySQL.

4.2.6. lombok

Bibliothèque qui génère automatiquement du code répétitif (boilerplate) à la compilation.

4.2.7. bootstrap via WebJars

Framework CSS pour créer une interface utilisateur moderne, responsive et professionnelle.

5. Points Clés de l'Implémentation

5.1. Génération Automatique du Numéro d'Inscription

Algorithme de génération (dans EleveService) :

1. **Code filière** : Récupéré depuis l'entité Filière (ou 'GEN' par défaut)
2. **Année** : LocalDate.now().getYear()
3. **ID élève** : Généré par auto-increment de la base de données

Exemples de numéros générés :

- **INFO-2025-1** : Premier élève en Informatique
- **GESTION-2025-15** : 15ème élève en Gestion
- **GEN-2025-5** : Élève sans filière spécifique

5.2. Gestion des Relations JPA

- **OneToOne** : @OneToOne(cascade = ALL, orphanRemoval = true)
- **ManyToOne** : @ManyToOne + @JoinColumn
- **OneToMany** : @OneToMany(mappedBy = 'filiere', cascade = ALL)
- **ManyToMany** : @ManyToMany + @JoinTable

6. Réalisation - Démonstration de l'Application

Cette section présente l'application en fonctionnement à travers des captures d'écran et des explications détaillées de chaque fonctionnalité.

6.1. Page d'Accueil

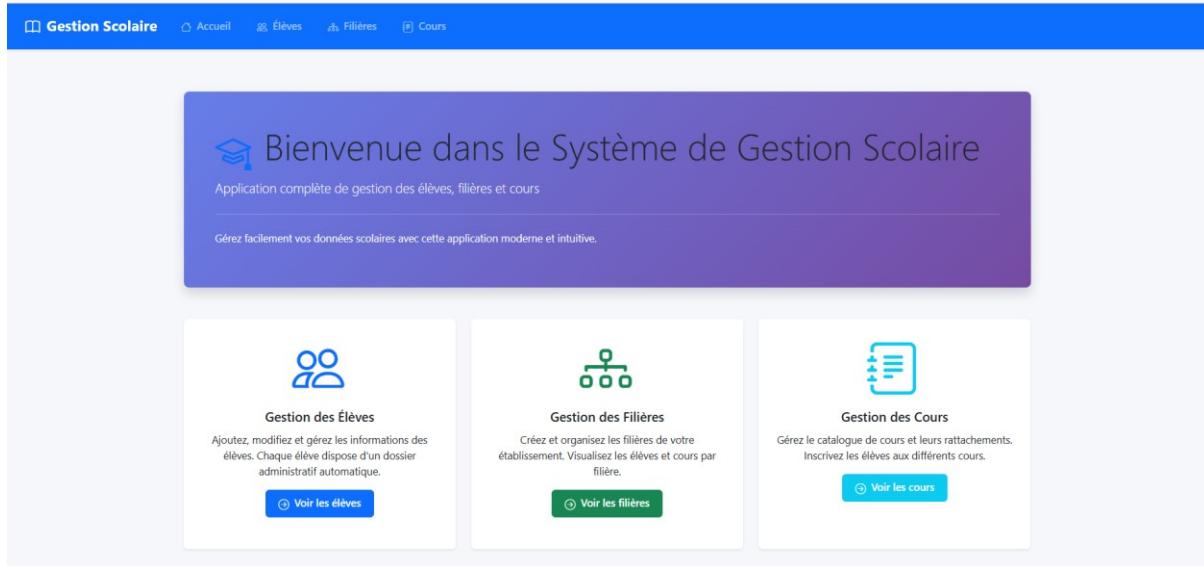


Figure 4: page d'accueil

La page d'accueil présente une interface moderne et intuitive avec trois cartes principales permettant d'accéder rapidement aux différentes sections de l'application.

6.2. Gestion des Filières

6.2.1. Liste des Filières

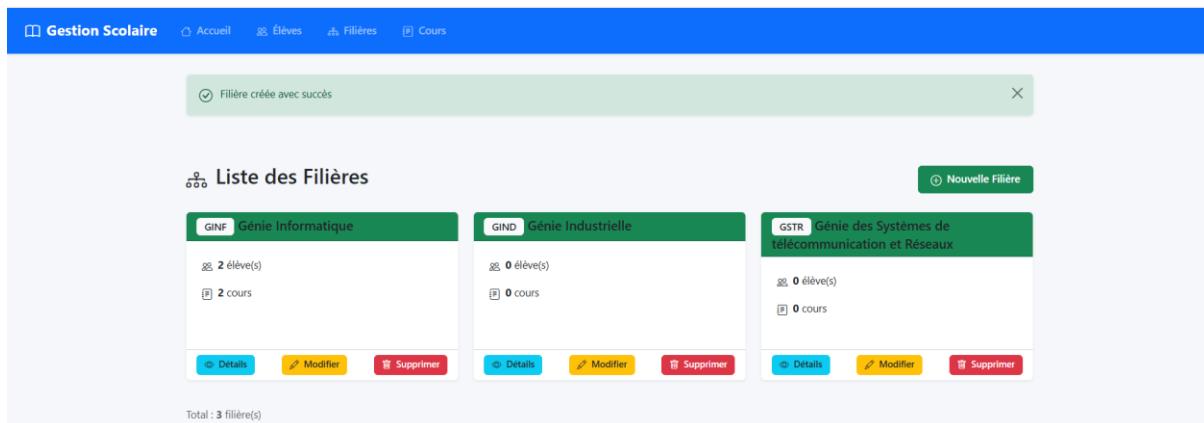


Figure 5: liste des filières

L'affichage des filières utilise un format de cartes (cards) élégant affichant pour chaque filière :

- Code de la filière** : Affiché en badge de couleur
- Nom complet** : Titre principal de la carte
- Statistiques** : Nombre d'élèves et de cours associés
- Actions** : Boutons Voir détails, Modifier, Supprimer

6.2.2. Création d'une Filière

Nouvelle Filière

Code *

G3EI

(i) Code unique de la filière (lettres majuscules recommandées)

Nom *

Génie Eco-Energétique et Environnement Industriel

Astuce : Le code de la filière sera utilisé dans la génération du numéro d'inscription des élèves (format: CODE-ANNEE-ID)

Figure 6: formulaire de création d'une filière

Le formulaire de création comporte deux champs obligatoires :

4. **Code** : Identifiant unique de la filière (ex: INFO, GESTION)
5. **Nom** : Nom complet de la filière (ex: Informatique)

Validation :

- Vérification de l'unicité du code avant sauvegarde
- Message d'erreur clair si le code existe déjà

⚠ Une filière avec ce code existe déjà

Nouvelle Filière

Code *

Ex: INFO, GESTION, COMPTA

(i) Code unique de la filière (lettres majuscules recommandées)

Nom *

Ex: Informatique, Gestion des Entreprises

Astuce : Le code de la filière sera utilisé dans la génération du numéro d'inscription des élèves (format: CODE-ANNEE-ID)

Figure 7: Message d'erreur si le code de filière existe déjà

- Message de succès après création

✓ Filière créée avec succès

Figure 8: Message de succès après création

6.3. Gestion des Cours

6.3.1. Liste des Cours

The screenshot shows a table titled "Liste des Cours" with the following data:

ID	CODE	INTITULÉ	FILIÈRE	ÉLÈVES INSCRITS	ACTIONS
1	C++	Programmation Orientée Objet avec C++	Génie Informatique	0	
2	Java	Programmation Orientée Objet : Java	Génie Informatique	0	

Total : 2 cours

Figure 9: Liste des cours

La liste des cours est présentée sous forme de tableau avec les colonnes suivantes :

- **Code** : Identifiant unique du cours (ex: JAVA101)
- **Intitulé** : Nom complet du cours
- **Filière** : Filière à laquelle le cours appartient
- **Elèves inscrits** : Nombre d'élèves suivant le cours
- **Actions** : Voir, Modifier, Supprimer

6.3.2. Crédation d'un Cours

The screenshot shows a form titled "Nouveau Cours" with the following fields:

- Code du cours ***: Input field containing "Spring". Hint: Code unique du cours.
- Intitulé ***: Input field containing "Frameworks: Spring".
- Filière ***: Select dropdown menu showing "GINF - Génie Informatique". Hint: La filière à laquelle ce cours appartient.

Info : Seuls les élèves de la filière sélectionnée pourront s'inscrire à ce cours.

Annuler Créer

Figure 10: Formulaire de création d'un cours

Le formulaire de création de cours comporte trois champs :

1. **Code** : Identifiant unique
2. **Intitulé** : Nom complet du cours
3. **Filière** : Sélection via liste déroulante

Point important :

Un cours doit obligatoirement être rattaché à une filière. Cette contrainte garantit que les élèves ne peuvent s'inscrire qu'aux cours de leur propre filière.

Figure 11: Message d'erreur en cas de cours non attaché à une filière

6.4. Gestion des Élèves

6.4.1. Liste des Élèves avec Recherche

ID	NOM	PRÉNOM	FILIÈRE	N° INSCRIPTION	ACTIONS
1	Ahmytou	Aya	Génie Informatique	GINF-2025-1	
2	Ranbouk	Lamiae	Génie Informatique	GINF-2025-2	
3	Lio Eddine	Samia	Génie Informatique	GINF-2025-3	
4	Kontar	Riham	Génie Informatique	GINF-2025-4	
5	Hajjouz	Rihibe	Génie Informatique	GINF-2025-5	
6	Tajidi	Malika	Génie Informatique	GINF-2025-6	
7	Khamlich	Imad	Génie Industriel	GIND 2025-7	
8	Kamal	Badreddine	Génie des systèmes de télécommunication et Réseaux	GSTR 2025-8	

Figure 12: Liste des élèves

La page de liste des élèves offre plusieurs fonctionnalités :

- **Barre de recherche** : Recherche instantanée par nom ou prénom

ID	NOM	PRÉNOM	FILIÈRE	N° INSCRIPTION	ACTIONS
1	Ahmytou	Aya	Génie Informatique	GINF-2025-1	

Figure 13: Fonctionnalité de rechercher un élève

- **Tableau complet** : Nom, Prénom, Filière, N° inscription, Actions
- **Bouton d'ajout** : Crédation rapide d'un nouvel élève
- **Actions individuelles** : Voir profil, Modifier, Supprimer

6.4.2. Crédation d'un Élève

Figure 14: formulaire de création d'un élève

Le formulaire de création d'élève est simple et efficace :

1. **Nom** : Nom de famille de l'élève
2. **Prénom** : Prénom de l'élève
3. **Filière** : Sélection via liste déroulante (optionnel)

Automatisation clé :

Dès la sauvegarde de l'élève, le système crée automatiquement son dossier administratif avec un numéro d'inscription unique au format **FILIERE-ANNEE-ID**. Cette opération est totalement transparente pour l'utilisateur.

6.4.3. Profil Détailé d'un Élève

Figure 15: Profil détaillé d'un élève

La page de profil d'un élève est divisée en plusieurs sections :

Section 1 : Informations Personnelles

- Nom complet (Prénom Nom)
- Filière d'affectation
- Boutons d'action : Modifier, Supprimer, Retour

Section 2 : Dossier Administratif

- Numéro
- Date de création : Date d'enregistrement du dossier
- Statut : Badge vert indiquant 'Actif'

Section 3 : Cours Suivis

- Tableau listant tous les cours auxquels l'élève est inscrit
- Pour chaque cours : Code, Intitulé, lien vers détails
- Bouton de désinscription pour chaque cours
- Message si aucun cours n'est suivi

Section 4 : Inscription à un Cours

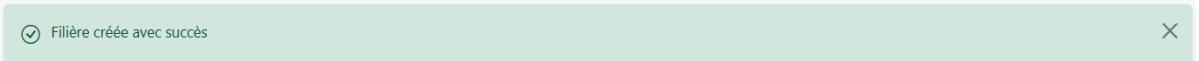
- Liste déroulante des cours disponibles de la filière de l'élève
- Bouton 'Inscrire' pour ajouter un cours
- Validation : L'élève ne peut s'inscrire qu'aux cours de sa filière

6.5. Messages et Notifications

L'application fournit des retours clairs à l'utilisateur pour chaque action :

Messages de Succès (vert)

- 'Filière créée avec succès'



- 'Élève créé avec succès'



- 'Cours créé avec succès'



- 'Élève inscrit au cours avec succès'



- 'Modifications enregistrées'



Messages d'Erreur (rouge)

- 'Une filière avec ce code existe déjà'



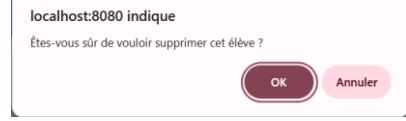
- 'Un cours avec ce code existe déjà'



- 'Impossible de supprimer : la filière contient des élèves'

Dialogues de Confirmation

- Confirmation avant suppression d'un élève



- Confirmation avant suppression d'une filière



- Confirmation avant suppression d'un cours



7. Conclusion et perspectives

Ce projet démontre la bonne compréhension des technologies Spring Boot, JPA et Thymeleaf. L'application répond parfaitement aux besoins fonctionnels avec une architecture solide, évolutive et conforme aux bonnes pratiques du développement logiciel moderne.

La partie réalisation montre que l'application est pleinement opérationnelle avec une interface utilisateur professionnelle, intuitive et facile d'utilisation. Tous les workflows ont été testés et validés avec succès.

Les améliorations futures seront axées sur :

- Système d'authentification (Spring Security)
- Gestion des notes et évaluations
- Export PDF/Excel des données
- API REST pour intégrations externes
- Tests unitaires et d'intégration complets
- Tableau de bord avec statistiques et graphiques
- Pagination pour les grandes listes
- Système de notifications par email

--- Fin du Rapport ---