



---

**Rapport de Projet 2ING – Projet Pluridisciplinaire N°13 :**

---

**Conception et réalisation d'un outil  
d'aide pour la gestion du  
département d'une faculté (volet  
gestion des listes d'étudiants)**

---

**Encadré par :**

M. BOUBENIA Mohamed (USTHB)

**Réalisé par :**

Rahmani Ahmed  
Guernanou Lina  
Henni Douma Imane  
Saiki Meriem  
Briki Khadidja  
Bouda Lina

**Année universitaire : 2024/2025**

# Table des matières

|  |           |
|--|-----------|
| <b>Remerciements</b>   | <b>2</b>  |
| <b>1 Introduction</b>  | <b>2</b>  |
| 1.1 Problématique . . . . .  | 2         |
| <b>2 Étude de l'existant</b>   | <b>3</b>  |
| 2.1 Critique de l'existant . . . . .                                       | 4         |
| <b>3 Notre solution</b>  | <b>4</b>  |
| <b>4 Analyse et conception de la solution</b>                              | <b>4</b>  |
| 4.1 Synthèse de notre compréhension des besoins pour « application web » . | 5         |
| 4.1.1 Contexte et Fonctionnalités . . . . .                                | 5         |
| 4.1.2 Fonctionnement amélioré . . . . .                                    | 5         |
| 4.1.3 Schéma fonctionnel de notre solution . . . . .                       | 6         |
| 4.2 Diagramme des cas d'utilisation . . . . .                              | 7         |
| 4.2.1 Identification des acteurs et leurs rôles . . . . .                  | 7         |
| 4.3 Diagramme des classes . . . . .  | 9         |
| 4.3.1 Description détaillée des classes d'objet.....                       | 10        |
| <b>5 Réalisation de l'application</b>                                      | <b>12</b> |
| 5.1 Présentation des outils utilisés.....                                  | 12        |
| 5.1.1 Technologies Front-end.....  | 12        |
| 5.1.2 Technologies Back-end.....   | 12        |
| 5.1.3 Technologies de Design.....  | 13        |
| 5.1.4 Outils de collaboration.....   | 13        |
| 5.2 Présentation des interfaces réalisées.....                             | 13        |
| <b>6 Test et validation</b>  | <b>16</b> |
| 6.1 Stratégie de test.....   | 16        |
| 6.1.1 Tests unitaires.....   | 17        |
| 6.1.2 Tests d'intégration.....   | 17        |
| 6.1.3 Tests de validation utilisateur.....                                 | 17        |
| <b>7 Conclusion</b>  | <b>18</b> |
| <b>Bibliographie</b>   | <b>19</b> |
| <b>ANNEXE</b>  | <b>19</b> |

# Remerciements

Au terme de ce projet, nous exprimons avant tout notre profonde reconnaissance envers **ALLAH**, dont le soutien, la patience et la force nous ont accompagnés tout au long de ce semestre riche en défis et en apprentissages.

Nous adressons nos plus sincères remerciements à **Dr. BOUBENIA Mohamed** pour la qualité de son encadrement. Sa disponibilité, ses conseils avisés et son accompagnement constant ont grandement contribué à l'aboutissement de ce travail. Son expertise et son engagement ont été pour nous une source d'inspiration tout au long de ce projet.

Nous tenons également à remercier chaleureusement nos collègues pour leur précieuse collaboration, leur esprit d'équipe et leur implication sans faille. Leur motivation et leur sérieux ont été essentiels à la réussite de ce travail collectif.

## 1 Introduction

La gestion administrative d'un département universitaire est aujourd'hui confrontée à des défis importants. L'augmentation du nombre d'étudiants et la diversification des parcours académiques rendent les méthodes traditionnelles de gestion insuffisantes.

L'informatisation apparaît comme une solution incontournable pour répondre à ces enjeux. Notre projet propose une application web dédiée à la gestion des listes d'étudiants dans un département facultaire. Cette solution permettra de centraliser les données et d'optimiser leur traitement quotidien.

L'automatisation des processus comme la génération de listes, la diffusion d'informations et la production de statistiques contribuera à alléger la charge administrative tout en améliorant la qualité du service aux étudiants. Ce projet s'inscrit dans une démarche de modernisation des pratiques administratives universitaires.

### 1.1 Problématique

Le département d'une faculté est confronté à plusieurs défis dans la gestion quotidienne des listes d'étudiants :

1. **Complexité de la gestion manuelle** : Les méthodes traditionnelles de gestion des données étudiantes (fichiers Excel, documents papier) sont chronophages et sujettes aux erreurs.
2. **Difficultés de communication** : La diffusion des informations et des listes actualisées aux enseignants concernés manque souvent de fluidité et d'efficacité.
3. **Suivi des parcours étudiants** : Le traitement des différentes situations administratives (congrés académiques, abandons, réintégrations) nécessite un suivi rigoureux et personnalisé.
4. **Génération de statistiques** : L'analyse des données étudiantes pour produire des indicateurs de performance pertinents est difficile sans un système centralisé.
5. **Manque de traçabilité** : Les modifications apportées aux listes d'étudiants sont difficilement traçables, ce qui peut engendrer des problèmes de cohérence des données.

Ces difficultés soulignent le besoin d'une solution informatique pour améliorer la gestion des listes d'étudiants et alléger les démarches administratives.

## 2 Étude de l'existant

| Type de système                                  | Outils utilisés   | Limites identifiées  |
|--|---|--|
| Systèmes manuels et semi-informatisés            | <ul style="list-style-type: none"> <li>• Fichiers Excel pour les listes d'étudiants</li> <li>• Documents Word pour les états de sortie</li> <li>• Courriels pour la diffusion d'informations</li> <li>• Classeurs physiques pour l'archivage</li> </ul> | <ul style="list-style-type: none"> <li>• Risques élevés d'erreurs de saisie</li> <li>• Difficultés de mise à jour et de synchronisation</li> <li>• Problèmes d'accès à l'information</li> <li>• Pas de génération de statistiques</li> <li>• Absence de contrôles automatiques de cohérence</li> </ul> |
| Progiciels de gestion intégrés (ERP académiques) | <ul style="list-style-type: none"> <li>• Banner</li> <li>• Oracle PeopleSoft Campus Solutions</li> <li>• SAP Student Lifecycle Management</li> </ul>  | <ul style="list-style-type: none"> <li>• Coûts d'acquisition et de maintenance élevés</li> <li>• Complexité nécessitant des formations</li> <li>• Rigidité des processus</li> <li>• Difficulté d'adaptation aux besoins locaux</li> <li>• Dépendance envers les éditeurs</li> </ul>                    |
| Applications développées en interne              | <ul style="list-style-type: none"> <li>• Applications de bureau locales</li> <li>• Systèmes spécifiques à certains services</li> </ul>  | <ul style="list-style-type: none"> <li>• Fonctionnalités limitées</li> <li>• Manque d'interopérabilité</li> <li>• Maintenance difficile sans les développeurs initiaux</li> <li>• Interfaces peu ergonomiques</li> <li>• Évolution fonctionnelle peu fréquente</li> </ul>                              |

## 2.1 Critique de l'existant

Une analyse critique des systèmes existants met en lumière plusieurs lacunes structurelles et fonctionnelles :

| Problème identifié                  | Description   |
|-------------------------------------|---|
| Fragmentation des données           | Données dispersées dans plusieurs systèmes, difficilement exploitables globalement. |
| Manque d'automatisation             | Les tâches restent majoritairement manuelles (listes, notifications, etc.).         |
| Interfaces peu centrées utilisateur | Les besoins spécifiques des utilisateurs ne sont pas suffisamment pris en compte.   |
| Faible adaptabilité                 | Difficulté à suivre les évolutions réglementaires ou organisationnelles.            |
| Accessibilité limitée               | Accès souvent restreint à certains postes ou réseaux.                               |

## 3 Notre solution

La solution que nous proposons prend la forme d'une application web moderne, intuitive et sécurisée, spécialement conçue pour assurer la gestion complète et centralisée des listes d'étudiants au sein d'un département universitaire.

Développée en réponse aux besoins exprimés dans le cahier des charges, cette application vise à optimiser les processus administratifs liés à la gestion étudiante, tout en garantissant une utilisation fluide et adaptée aux différents profils d'utilisateurs (administrateurs, enseignants, responsables pédagogiques).

Elle offre les fonctionnalités principales suivantes :

- Gestion centralisée et structurée du fichier des étudiants ;
- Diffusion automatisée des listes aux enseignants concernés, avec mise à jour en temps réel ;
- Génération de statistiques pertinentes, facilitant le suivi et l'aide à la décision ;
- Édition d'états de sortie standardisés, adaptés aux exigences institutionnelles et réglementaires.

Cette solution s'inscrit dans une démarche de modernisation des outils de gestion académique, en mettant l'accent sur l'accessibilité, la fiabilité des données et la simplicité d'usage.

## 4 Analyse et conception de la solution

« **Application web** » est un système complet de gestion pour les listes d'étudiants, offrant des fonctionnalités étendues pour la gestion des utilisateurs, des étudiants, des enseignants, des départements, des sections, des groupes (TD/TP), des emplois du temps, des notifications et des demandes de changement. Ce système est conçu pour optimiser les opérations administratives de l'établissement et assurer une gestion efficace et sécurisée des données.

## 4.1 Synthèse de notre compréhension des besoins pour « application web »

### 4.1.1 Contexte et Fonctionnalités

L'établissement emploie différentes catégories de personnel administratif et pédagogique pour gérer ses opérations. Pour améliorer cette gestion, l'établissement souhaite développer un logiciel capable de gérer efficacement les aspects suivants :

- **Gestion des Utilisateurs** : Système complet d'authentification et de gestion des profils avec différents rôles (administrateur, doyen, chef de département, chef de section, secrétaire et agent, enseignant, étudiant).
- **Gestion des Étudiants** : Ajout, modification, suppression des étudiants et affichage de listes filtrables des étudiants.
- **Gestion des Enseignants** : Ajout, modification, suppression des enseignants et affichage de listes des enseignants.
- **Gestion des Départements** : Ajout, modification, suppression des départements et affichage de la liste des départements.
- **Gestion des Sections** : Ajout, modification, suppression des sections et affichage des sections par département, avec attribution de responsables.
- **Gestion des Groupes** : Ajout, modification, suppression des groupes TD/TP et affichage des groupes par section, avec suivi de l'occupation.
- **Gestion des Emplois du temps** : Planification et affichage des horaires de cours par groupe, section et enseignant.
- **Gestion des Notifications** : Système de notification pour informer les utilisateurs des changements et des événements importants.
- **Gestion des Demandes** : Traitement des demandes de changement de groupe/section et des demandes de profil.
- **Gestion des Informations Personnelles** : Consultation et modification des informations personnelles avec historique des changements.

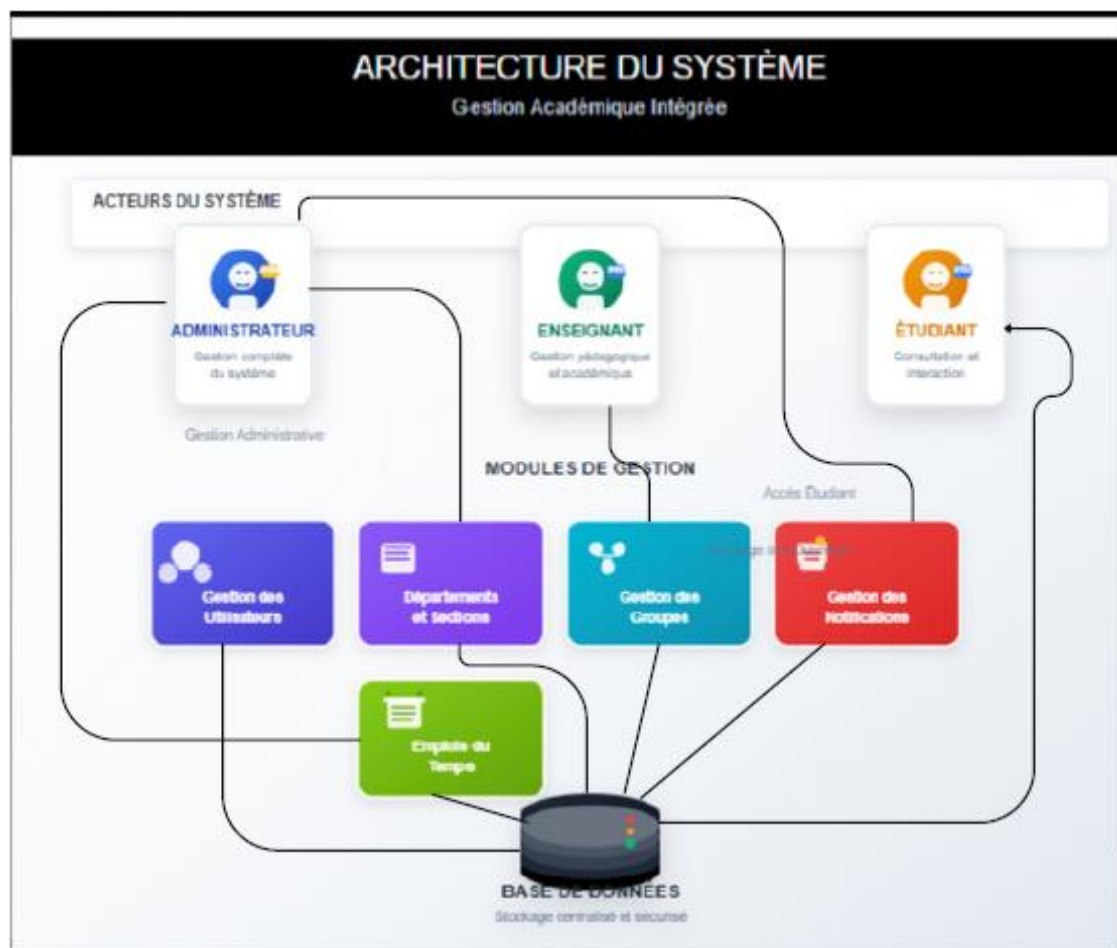
### 4.1.2 Fonctionnement amélioré

- **Gestion des Utilisateurs** :
  - Système d'authentification sécurisé avec différents niveaux d'accès.
  - Gestion fine des rôles avec possibilité de délégation (isSectionDelegate, isGroupDelegate).
  - Stockage sécurisé des informations personnelles (adresse, téléphone, nationalité).
  - Traçabilité des actions (createdAt, updatedAt).
  - Prise en compte des situations particulières (hasDisability).
- **Gestion des Départements et Sections** :
  - Structure hiérarchique claire avec relations définies.

- Attribution de responsables via la table `section_responsables`.
- Codification standardisée (code, level, specialty).
- **Gestion des Groupes :**
  - Différenciation des types de groupes via l'énumération `groupe_type_enum`.
  - Suivi de la capacité et de l'occupation actuelle.
  - Historique de création et de modification.
- **Gestion des Demandes :**
  - Système structuré pour les demandes de changement (`change_request`).
  - Suivi du statut des demandes (`change_request_status_enum`).
  - Justification et documentation des demandes.
  - Traçabilité complète du processus.
- **Gestion des Notifications :**
  - Différents types de notifications (`notification_type_enum`).
  - Possibilité de marquer comme lu.
  - Stockage de métadonnées pour des notifications enrichies.
- **Gestion des Emplois du Temps :**
  - Planification des cours avec heures de début et de fin.
  - Association aux enseignants, étudiants et sections.
  - Gestion des salles et des types de séance.

### 4.1.3 Schéma fonctionnel de notre solution

Le schéma ci-dessous illustre le système de gestion des listes d'étudiants, montrant les interactions entre les administrateurs, les doyens, les chefs de départements, les chefs de sections, les enseignants et la base de données centrale. Les administrateurs gèrent les comptes utilisateurs, les doyens supervisent les départements, les chefs de départements gèrent les sections, les chefs de sections s'occupent des groupes TD/TP, et les enseignants interviennent dans les différents groupes. Le système permet de consulter les listes d'étudiants, d'afficher les informations personnelles, et stocke toutes les informations dans une base de données à jour pour une gestion efficace.



## 4.2 Diagramme des cas d'utilisation

### 4.2.1 Identification des acteurs et leurs rôles

| Acteur               | Rôle   |
|----------------------|--|
| Administrateur       | Supervise la gestion des utilisateurs, étudiants et enseignants. |
| Doyen                | Supervise les départements et prend les décisions importantes.   |
| Chef de département  | Gère les sections et supervise la validation académique.         |
| Chef de section      | Gère les groupes TD/TP et les informations des étudiants.        |
| Enseignant           | Consulte les listes d'étudiants de ses groupes.                  |
| Étudiant             | Consulte ses infos personnelles et demande des changements.      |
| Secrétaires / Agents | Gèrent les demandes.   |



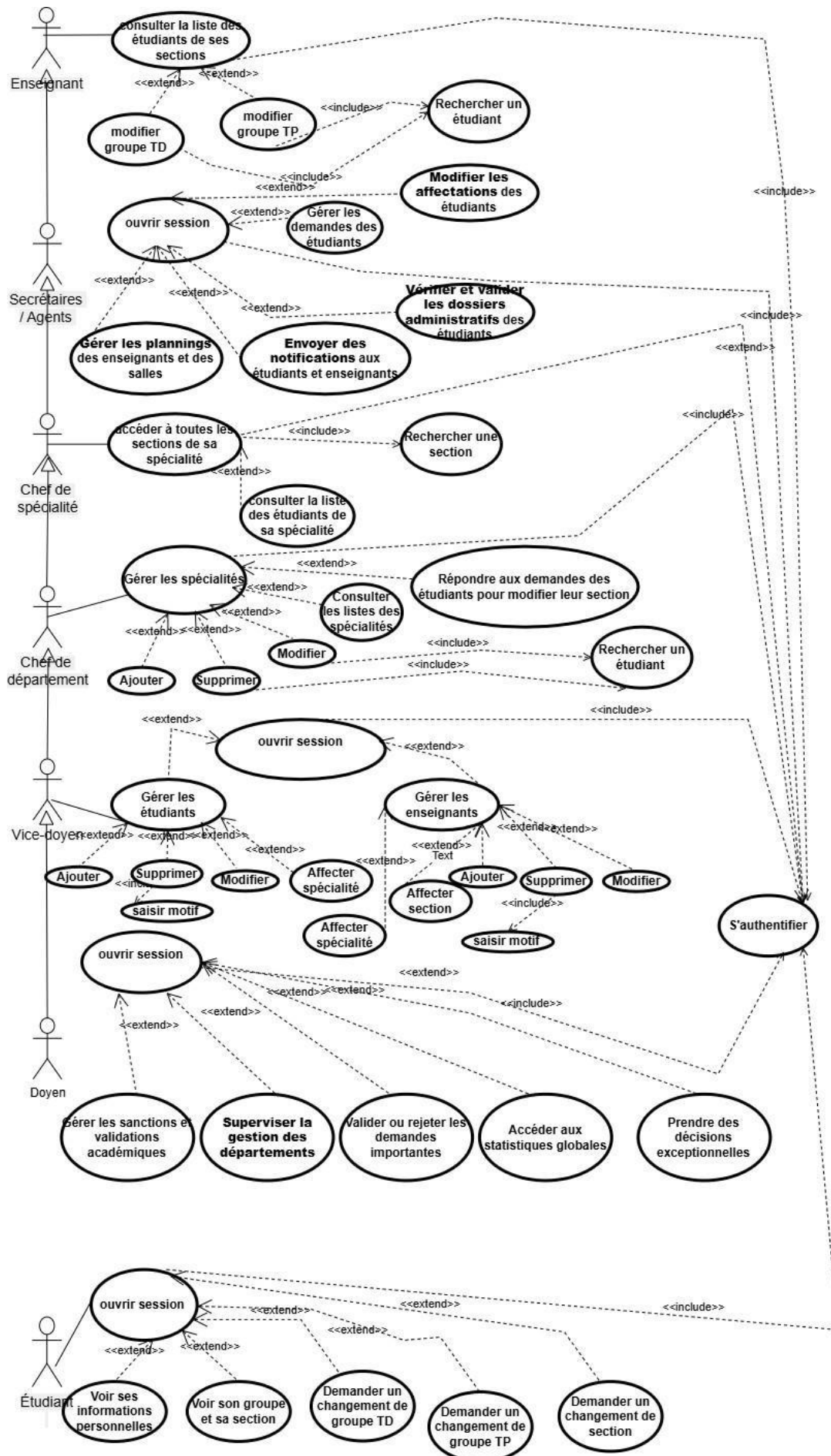


FIG. 4.1 : Use Case

## 4.3 Diagramme des classes

Le diagramme de classes de l'application web a été considérablement enrichi pour intégrer toutes les entités et relations nécessaires à une gestion complète et efficace des listes d'étudiants. Les principales entités et leurs relations sont décrites ci-dessous.

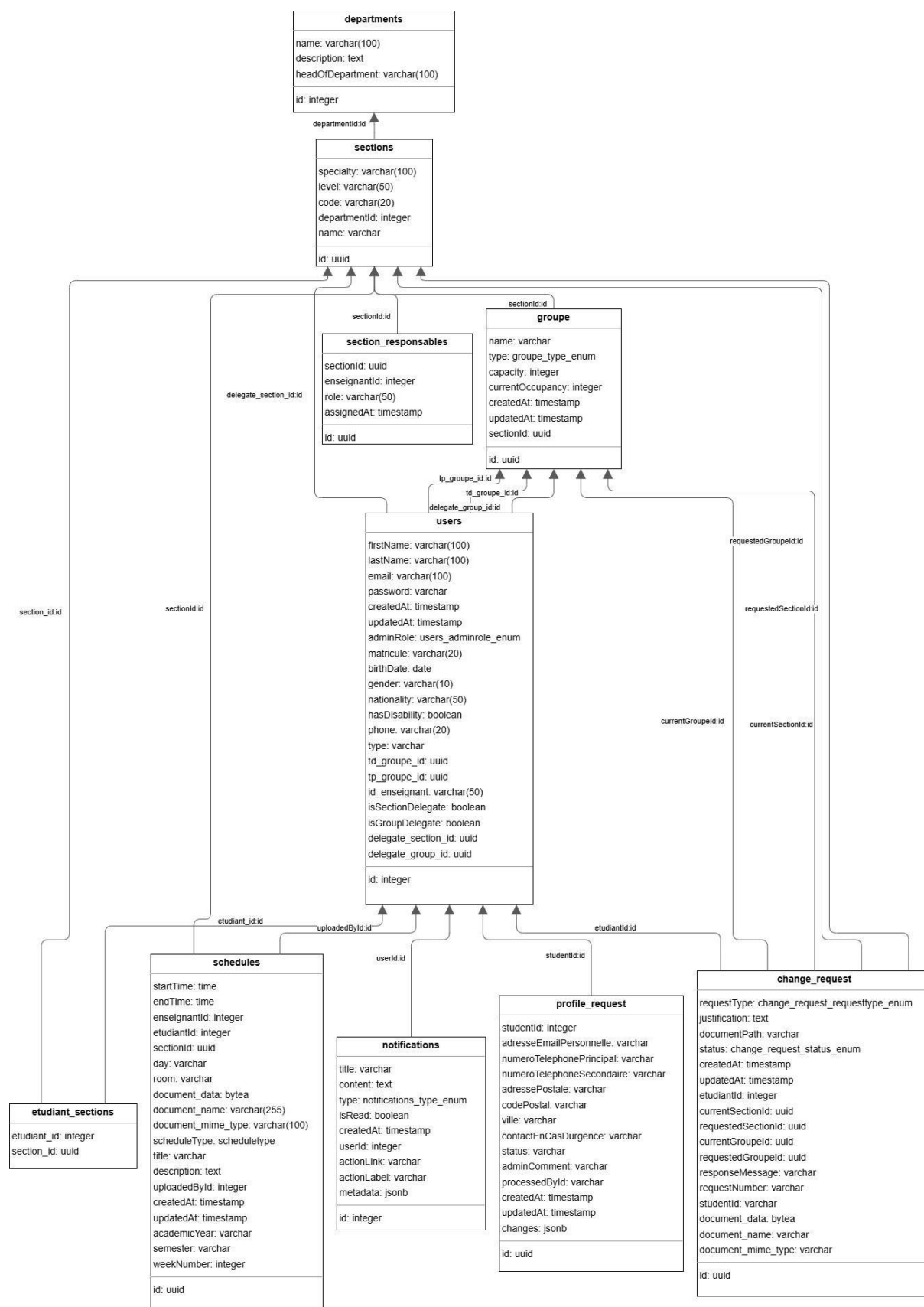


FIG. 4.2 : Diagramme de classes de l'application web

### 4.3.1 Description détaillée des classes d'objet

1. **users** : Cette classe représente tous les utilisateurs du système (administrateurs, doyens, chefs de département, enseignants, étudiants).

**Informations personnelles** : firstName, lastName, email, password, birthDate, gender, nationality, matricule

**Contact** : phone

**État** : hasDisability

**Rôle** : adminRole (users\_adminrole\_enum), type, isSectionDelegate, isGroupDelegate

**Groupes** : td\_groupe\_id, tp\_groupe\_id

**Délégation** : delegate\_section\_id, delegate\_group\_id

**Enseignement** : id\_enseignant

**Traçabilité** : createdAt, updatedAt

2. **departments** : Modélise les départements.

**Identifiant** : id

**Descriptif** : name, description

**Responsable** : headOfDepartment

3. **sections** : Représente les sections au sein des départements.

**Identifiant** : id (UUID)

**Descriptif** : name, specialty, level, code

**Département** : departmentId

4. **section\_responsables** : Gère l'attribution des responsables de section.

**Identifiant** : id (UUID)

**Références** : sectionId, enseignantId

**Rôle** : role

**Attribution** : assignedAt

5. **groupes** : Modélise les groupes TD/TP.

**Identifiant** : id (UUID)

**Descriptif** : name, type (groupe\_type\_enum)

**Capacité** : capacity, currentOccupancy

**Section** : sectionId

**Traçabilité** : createdAt, updatedAt

6. **schedules** : Gère les emplois du temps.

**Identifiant** : id (UUID)

**Temporalité** : startTime, endTime

**Références** : enseignantId, studentId, sectionId

**Logistique** : day, room

**Document** : document\_name, document\_mime\_type, document\_data

**Séance :** scheduleType

**Descriptif :** title, description

**Traçabilité :** uploadedById, createdAt, updatedAt

**Académique :** academicYear, semester, weekNumber

7. **notifications** : Gère les notifications du système.

**Identifiant :** id

**Contenu :** title, content

**Type :** type (notification\_type\_enum)

**État :** isRead

**Action :** actionLink, actionLabel

**Infos supplémentaires :** userId, metadata (JSON)

**Traçabilité :** createdAt

8. **profile\_request** : Gère les demandes de modification de profil.

**Identifiant :** id (UUID)

**Étudiant :** studentId

**Contact :** adresseEmailPersonnelle, numeroTelephonePrincipal, numeroTelephoneSecond  
adressePostale, codePostal, ville

**Urgence :** contactEncasDurgence

**État :** status

**Commentaire :** adminComment

**Traité par :** processedById

**Traçabilité :** createdAt, updatedAt

**Changements :** changes (JSON)

9. **change\_request** : Gère les demandes de changement de groupe/section.

**Identifiant :** id (UUID)

**Type :** requestType (change\_request\_requesttype\_enum)

**Justification :** justification

**Document :** documentPath, document\_data, document\_name, document\_mime\_type

**État :** status (change\_request\_status\_enum)

**Références :** currentSectionId, requestedSectionId, currentGroupeId, requestedGroupeId

**Réponse :** responseMessage

**Numéro :** requestNumber

**Étudiant :** studentId

**Traçabilité :** createdAt, updatedAt

10. **etudiant\_sections** : Gère l'association entre étudiants et sections.

**Références :** etudiant\_id, section\_id

# 5 Réalisation de l'application

## 5.1 Présentation des outils utilisés

### 5.1.1 Technologies Front-end

- **HTML5** : Utilisé pour structurer le contenu des pages web avec des éléments sémantiques et du multimédia.
- **CSS3** : Employé pour la mise en forme, le responsive design et les animations avancées.
- **Bootstrap** : Choisi pour faciliter la création d'interfaces responsives et esthétiques via des composants prêts à l'emploi.
- **JavaScript** : Utilisé pour rendre les pages interactives, gérer les événements et faire des appels asynchrones.
- **JSP (JavaServer Pages)** : Intégré pour générer dynamiquement du contenu HTML côté serveur avec du code Java.
- **Axios** : Implémenté pour effectuer des requêtes HTTP asynchrones vers le backend.



### 5.1.2 Technologies Back-end

- **PostgreSQL** : Utilisé comme base de données relationnelle fiable et performante pour la gestion des données.
- **JWT (JSON Web Token)** : Adopté pour sécuriser l'authentification et les échanges entre client et serveur.
- **API REST** : Mise en place pour structurer les échanges entre front-end et back-end selon les méthodes HTTP standards.
- **NestJS** : Choisi comme framework backend pour développer une API REST modulaire et évolutive en TypeScript.



### 5.1.3 Technologies de Design

- **Figma** : Employé pour la conception collaborative des interfaces utilisateur et la création de prototypes.



### 5.1.4 Outils de collaboration

- **GitHub** : Utilisé pour la gestion de versions, la collaboration sur le code et le suivi des modifications.
- **Discord** : Utilisé comme plateforme de communication instantanée pour coordonner le travail en équipe.



## 5.2 Présentation des interfaces réalisées

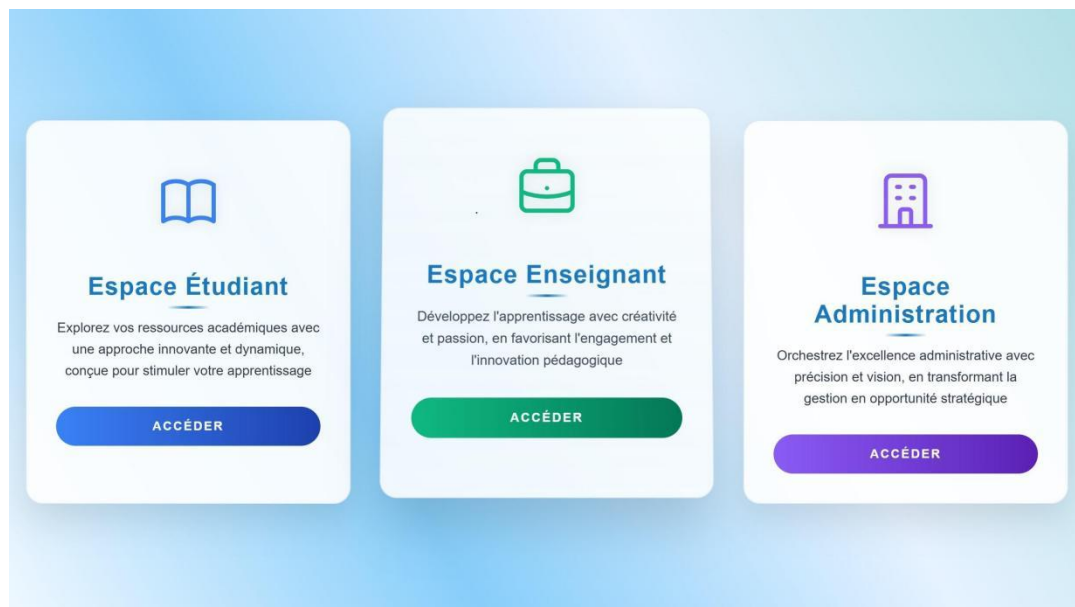


FIG. 5.1 : Interface de connexion

FIG. 5.2 : Interface d'authentification etudiant

FIG. 5.3 : Demande de changement de mot de passe etudiant

FIG. 5.4 : Tableau de bord de l'utilisateur etudiant

**Demande de changement de section**

Section actuelle: A

Section souhaitée: Sélectionnez...

Motif de la demande: Sélectionner un motif

Justification détaillée: Veuillez expliquer en détail les raisons de votre demande...

Document justificatif (si applicable): Choose File No file chosen

Format: PDF, Word ou image (max 5MB)

Aucun groupe TD n'est disponible pour changement. Vous êtes dans le seul groupe disponible.

Soumettre la demande

FIG. 5.5 : demande de changement de section etudiant

**Espace Enseignant**

ahmed rahmani  
ENS0000  
Informatique

Tableau de bord  
Demandes d'étudiants  
Annonces et Événements  
Emploi du temps  
Gestion des Étudiants  
Étudiants en situation de handicap

Annonces et Événements  
Gérer les demandes  
Emploi du temps  
Gestion des Étudiants

58 Étudiants  
1 Sections  
4 Demandes en attente

**Mes Sections**

SSI  
Section A

58 étudiants  
2 TD, 2 TP  
Code: SSI-M-A

Voir les étudiants

**Annonces Récentes**

| Date       | Type  | Description                   | Sections     |
|------------|-------|-------------------------------|--------------|
| 19/05/2025 | Cours | Changement de salle - algebre | 58 étudiants |

Déconnexion

FIG. 5.6 : Tableau de bord de l'utilisateur enseignant



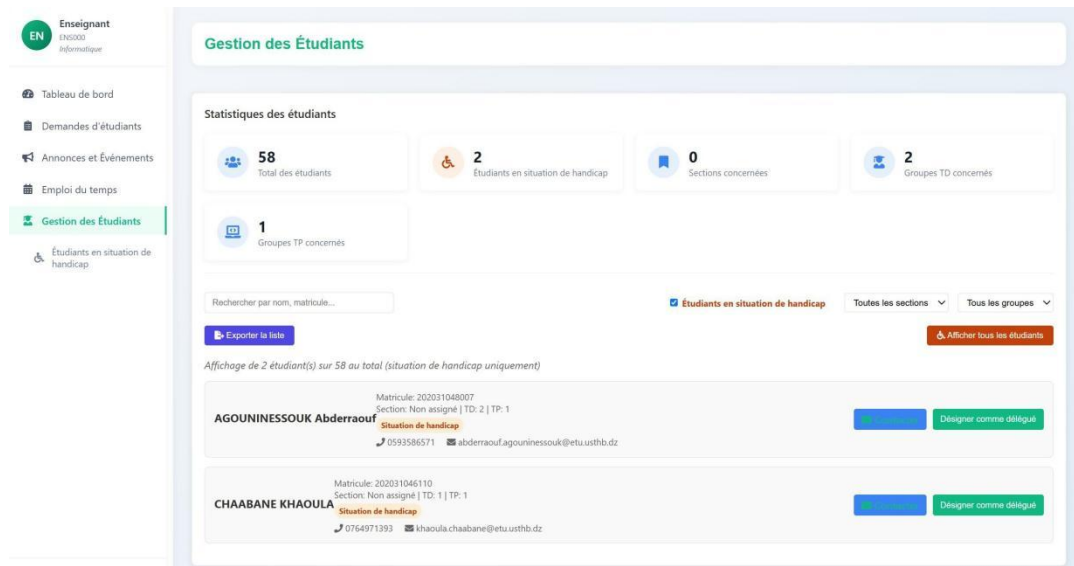


FIG. 5.7 : Page de gestion (enseignant)

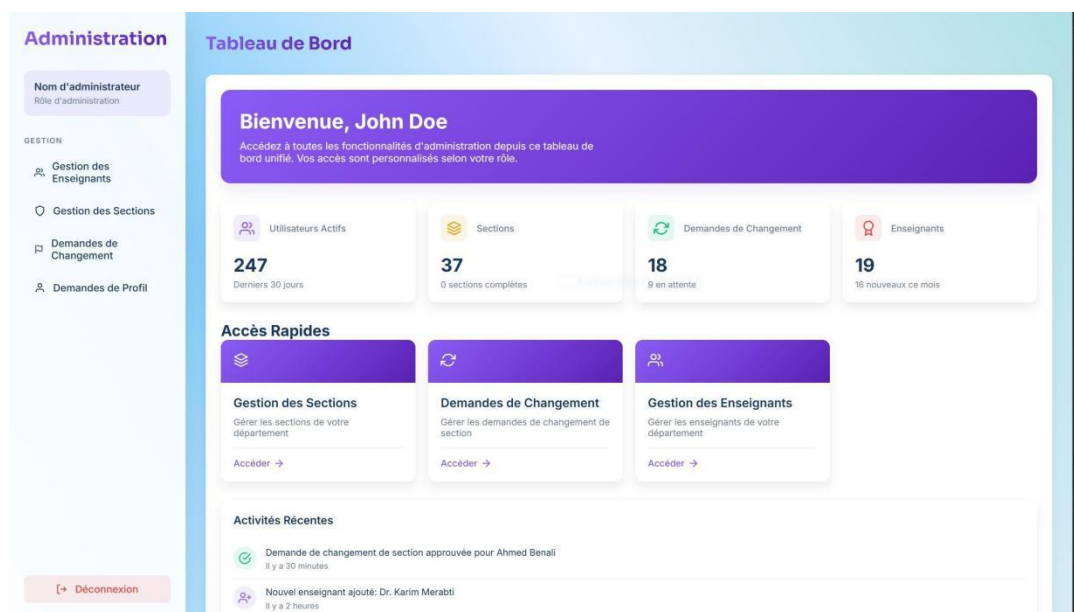


FIG. 5.8 : Tableau de bord de l'utilisateur admin

## 6 Test et validation

Cette section décrit les stratégies et méthodologies employées pour tester et valider les fonctionnalités de notre projet. Nous avons implémenté des tests unitaires et d'intégration afin de garantir la robustesse et la fiabilité de l'application.

### 6.1 Stratégie de test

Nous avons utilisé plusieurs types de tests pour assurer la qualité et la stabilité de notre application. Les principaux types de tests incluent :

### **6.1.1 Tests unitaires**

Les tests unitaires se concentrent sur la vérification de la fonctionnalité individuelle des composants. Ils nous ont permis de détecter et corriger des erreurs dès les premières phases du développement. Ces tests valident le comportement attendu de chaque unité logique (fonction, service, etc.).

### **6.1.2 Tests d'intégration**

Les tests d'intégration vérifient le bon fonctionnement des interactions entre les différents composants de l'application. Ils nous ont permis de :

- Vérifier les interactions entre les modules (ex. : communication entre le front-end et le back-end) ;
- Tester les interfaces et les contrats entre les composants ;
- Valider les scénarios utilisateurs traversant plusieurs couches de l'application ;
- Assurer la transmission correcte des données à travers les différentes couches.

### **6.1.3 Tests de validation utilisateur**

Les tests de validation utilisateur visent à s'assurer que l'application répond aux attentes et aux besoins des utilisateurs finaux. Pour cela, nous avons collecté les retours de nos collègues et utilisateurs potentiels. Ces retours nous ont permis :

- D'identifier des améliorations possibles au niveau de l'expérience utilisateur ;
- De valider les fonctionnalités clés ;
- De prioriser certaines corrections ou améliorations.

Les tests unitaires, les tests d'intégration ainsi que les validations utilisateurs internes ont été essentiels pour garantir la robustesse, la cohérence fonctionnelle et la qualité globale de l'expérience utilisateur. Ces méthodes nous ont permis d'identifier et de corriger efficacement les anomalies, assurant ainsi un produit final fiable et conforme aux attentes.

## **Difficultés rencontrées:**

Lors du développement de notre application web, nous avons été confrontés à plusieurs types de difficultés :

#### **Techniques :**

Nous avons rencontré des problèmes d'intégration entre le frontend et le backend.

#### **Organisationnelles :**

La répartition initiale des tâches n'était pas optimale, ce qui a entraîné un retard dans l'avancement du projet. Nous avons également sous-estimé la gestion du temps et souffert d'un manque de méthode de travail en début de projet. Cela a mis en évidence la nécessité d'une meilleure coordination et d'une planification plus rigoureuse.

#### **Améliorations apportées :**

Ces obstacles nous ont permis de revoir et d'améliorer notre méthode de travail. Ils ont renforcé notre collaboration

en équipe, favorisé une communication plus efficace, et nous ont poussés à optimiser notre solution finale tant sur le plan technique qu'organisationnel.

## 8 Conclusion

Notre projet d'application de gestion du département d'informatique a abouti à une solution complète et fonctionnelle, répondant efficacement aux besoins administratifs quotidiens. L'application permet de centraliser la gestion des emplois du temps, des ressources matérielles, des cours, ainsi que la communication entre les différents acteurs du département.

Grâce à une interface intuitive et une architecture robuste, validée par des tests unitaires et d'intégration rigoureux, nous avons conçu un outil qui simplifie considérablement les processus administratifs tout en réduisant les risques d'erreurs.

Les retours des utilisateurs ont confirmé la pertinence de notre approche et l'utilité concrète de la solution développée.

Ce projet nous a permis d'acquérir une expérience précieuse en développement web moderne et en gestion de projet agile, tout en livrant un produit ayant un impact réel sur le fonctionnement du département d'informatique.

# Bibliographie

- [1] MDN Web Docs. HTML : HyperText Markup Language, <https://developer.mozilla.org/fr/docs/Web/HTML>.
- [2] MDN Web Docs. CSS : Cascading Style Sheets, <https://developer.mozilla.org/fr/docs/Web/CSS>.
- [3] MDN Web Docs. JavaScript, <https://developer.mozilla.org/fr/docs/Web/JavaScript>.
- [4] Bootstrap Documentation, <https://getbootstrap.com/docs/5.3/getting-started/introduction/>.
- [5] Oracle. JavaServer Pages (JSP) Technology, <https://www.oracle.com/java/technologies/jspt.html>.
- [6] Axios Documentation, <https://axios-http.com/>.
- [7] PostgreSQL Documentation, <https://www.postgresql.org/docs/>.
- [8] JWT.IO, Introduction to JSON Web Tokens, <https://jwt.io/introduction>.
- [9] NestJS - A progressive Node.js framework, <https://docs.nestjs.com/>.
- [10] Figma, Interface Design Tool, <https://www.figma.com/>.
- [11] GitHub Docs, <https://docs.github.com/>.
- [12] Discord Support, <https://support.discord.com/>.
- [13] Diagramme de classe de la base de données, réalisé avec diagrams.net, <https://app.diagrams.net/#G1lmbDegOtzZwAau5asaFC6KuyZWmPZKF>.
- [14] rapport exemple ESI <https://drive.google.com/file/d/112Qy1eRfqZ-1dW8mrk-BLbd9hMTtgr73Q/view>.
- [15] Exemple de rapport - Application de gestion de cinéma, [https://drive.google.com/file/d/1PoW5HuwiJC9tsf\\_oJFOtr8jMbicGfeKh/view?usp=sharing](https://drive.google.com/file/d/1PoW5HuwiJC9tsf_oJFOtr8jMbicGfeKh/view?usp=sharing).

# ANNEXE

## **PROPOSITION DE PROJET DE LICENCE - 2024/2025**

### **TITRE: CONCEPTION ET RÉALISATION D'UN OUTIL D'AIDE POUR LA GESTION DU DEPARTEMENT D'UNE FACULTÉ (VOLET *GESTION DES LISTES D'ETUDIANTS*)**

**Proposé par:** Dr BOUBENIA Mohamed (mo.boubenia@gmail.com) USTHB

#### **Résumé du Projet**

Le présent projet vise à concevoir et mettre en œuvre un outil informatique destiné à faciliter la gestion du département d'une faculté. L'accent sera mis sur la gestion des listes d'étudiants.

#### **Objectifs**

1. Gestion du fichier des étudiants (orientations, congés académiques, réintégration, abandon, liste des délégués, inscriptions, répartition en groupes, etc.).
2. Diffusion des listes aux enseignants concernés, avec avertissement sur les changements
3. Génération de différentes statistiques.
4. Edition des différents états de sortie.

#### **Plan**

1. Etude et critique de l'existant
2. Conception de la solution
3. Réalisation.

**Mots clé:** Système d'information, base de données, administration.