

# **Cahier des Charges du Projet : Application d'Archivage des Relevés de Notes des Étudiants de la Faculté des Sciences**

## **1. Introduction**

**Objet :** Développer une application web permettant l'archivage et la consultation des relevés de notes des étudiants de la Faculté des Sciences.

### **Périmètre :**

- Gestion des utilisateurs (administrateurs, enseignants, étudiants)
- Archivage et gestion des relevés de notes
- Consultation des relevés de notes par les étudiants et les enseignants

## **2. Description des Fonctionnalités**

### **2.1 Gestion des Utilisateurs**

- **Connexion et Authentification :**
  - Les utilisateurs doivent pouvoir se connecter avec un email et un mot de passe.
  - Trois rôles principaux : administrateur, enseignant, étudiant.
- **Gestion des Rôles :**
  - Les administrateurs peuvent gérer les comptes des enseignants et des étudiants.
  - Les enseignants peuvent ajouter et consulter les relevés de notes.
  - Les étudiants peuvent consulter leurs propres relevés de notes.

### **2.2 Archivage et Gestion des Relevés de Notes**

- **Ajout de Relevés de Notes :**
  - Les enseignants peuvent ajouter des relevés de notes pour leurs étudiants.
  - Chaque relevé de notes doit inclure les informations suivantes : étudiant, matière, session, note, date.
- **Modification et Suppression de Relevés de Notes :**
  - Les enseignants peuvent modifier ou supprimer les relevés de notes existants.
- **Consultation des Relevés de Notes :**
  - Les enseignants et les étudiants peuvent consulter les relevés de notes archivés.

### **2.3 Consultation des Relevés de Notes**

- **Accès aux Relevés :**
  - Les étudiants peuvent consulter leurs relevés de notes via une interface dédiée.
  - Les enseignants peuvent consulter les relevés de notes des étudiants de leurs matières.

## **3. Technologies Utilisées**

- **Backend :**
  - PHP 8
  - Base de données MySQL

- **Frontend :**
  - HTML5
  - CSS3
- **Frameworks et Bibliothèques :**
  - Bootstrap 4 (pour le style)
  - PDO (pour l'interaction avec la base de données)

## 4. Exigences Fonctionnelles

### 4.1 Interface Utilisateur

- **Formulaire de Connexion :**
  - Champs : Email, Mot de passe
  - Bouton de connexion
- **Tableau de Bord Administrateur :**
  - Gestion des utilisateurs (ajout, modification, suppression)
  - Consultation des relevés de notes
- **Tableau de Bord Enseignant :**
  - Ajout, modification et suppression des relevés de notes
  - Consultation des relevés de notes des étudiants
- **Interface Étudiant :**
  - Consultation des relevés de notes personnels

### 4.2 Gestion des Données

- **Base de Données :**
  - Table `users` : pour stocker les informations des utilisateurs (id, username, email, password, role, created\_at)
  - Table `enseignants` : pour stocker les informations des enseignants (matricule\_enseignant, nom, prenom, date\_naissance, created\_at)
  - Table `etudiants` : pour stocker les informations des étudiants (matricule\_etudiant, nom, prenom, date\_naissance, created\_at)
  - Table `sessions` : pour stocker les informations des étudiants (session, date\_debut, date\_fin, date\_naissance, created\_at)
  - Table `matieres` : pour stocker les informations des matières (code\_matiere, nom\_matiere, credit, created\_at)
  - Table `releves` : pour stocker les relevés de notes (id\_releve, matricule\_etudiant, code\_matiere, note, session, semestre, departement, annee, matricule\_enseignant, created\_at)

## 5. Exigences Non Fonctionnelles

- **Sécurité :**
  - Protection des mots de passe via le hachage (bcrypt).

- Vérification des autorisations d'accès pour les différentes actions.
- **Performance :**
  - Temps de réponse acceptable pour les actions de gestion et de consultation.
  - Optimisation des requêtes SQL pour éviter les lenteurs.
- **Accessibilité :**
  - Interface utilisateur responsive pour être utilisable sur différents appareils (ordinateurs, tablettes, smartphones).

## 6. Diagrammes UML

### 6.1 Cas d'Utilisation

- **Acteurs :**
  - Administrateur
  - Enseignant
  - Étudiant
- **Cas d'Utilisation :**
  - Connexion
  - Gestion des Utilisateurs (Administrateur)
  - Gestion des Relevés de Notes (Administrateur, Enseignant)
  - Consultation des Relevés de Notes (Administrateur, Enseignant, Étudiant)

### 6.2 Diagramme de Séquence

- **Connexion :**
  - L'utilisateur entre ses informations de connexion.
  - Le système vérifie les informations et authentifie l'utilisateur.
- **Gestion des Utilisateurs :**
  - L'administrateur sélectionne l'action (ajouter, modifier, supprimer).
  - Le système effectue l'action choisie et met à jour la base de données.
- **Gestion des Relevés de Notes :**
  - L'enseignant sélectionne l'action (ajouter, modifier, supprimer).
  - Le système effectue l'action choisie et met à jour la base de données.
- **Consultation des Relevés de Notes :**
  - L'utilisateur demande la liste des relevés de notes.
  - Le système récupère les données et les affiche à l'utilisateur.

## 7. Déroulement du Projet

### 7.1 Phases du Projet

- **Analyse et Conception :**
  - Définition des exigences
  - Conception de la base de données et des interfaces utilisateur
- **Développement :**

- Implémentation des fonctionnalités backend et frontend
- Tests unitaires et d'intégration
- **Tests et Validation :**
  - Tests fonctionnels et non fonctionnels
  - Validation des exigences
- **Déploiement et Maintenance :**
  - Déploiement du système sur un serveur de production
  - Maintenance et mises à jour régulières

## 7.2 Planning

- **Semaine 1-2 :** Analyse et Conception
- **Semaine 3-6 :** Développement
- **Semaine 7-8 :** Tests et Validation

## 8. Conclusion

Ce cahier des charges présente une vue d'ensemble des objectifs, des fonctionnalités et des exigences du projet d'application d'archivage des relevés de notes des étudiants de la Faculté des Sciences. Le respect des spécifications techniques et des délais assurera la réalisation d'un système performant et adapté aux besoins des utilisateurs.