

Cahier des Charges

Application de Gestion de Cours et d'Évaluations pour une École

➤ Identification du problème à résoudre

1.1 Contexte

Les établissements scolaires font face à des défis croissants dans la gestion administrative des cours, des emplois du temps, des évaluations et du suivi des élèves. Les méthodes traditionnelles (papier, tableurs épars) entraînent :

- Une perte de temps considérable pour le personnel administratif et enseignant
- Des risques d'erreurs dans la saisie et le traitement des données
- Une difficulté à centraliser et à accéder rapidement aux informations
- Un manque de communication efficace entre les différents acteurs (administration, enseignants, élèves, parents)

1.2 Problématiques spécifiques

- Planification complexe des cours et gestion des salles
- Suivi difficile des évaluations et des résultats des élèves
- Communication fragmentée entre l'école, les enseignants, les élèves et les parents
- Difficulté à produire des rapports et analyses sur les performances

➤ Définition des utilisateurs

2.1 Profils d'utilisateurs

Enseignants

- Besoins : Gestion des cours, création d'évaluations, notation, suivi des élèves

Élèves

- Besoins : Consultation des cours, accès aux supports, soumission des travaux, consultation des notes

➤ **Fonctionnalités principales**

Authentification des utilisateurs

- Connexion de base pour les enseignants et les étudiants (accès basé sur les rôles)

Gestion des cours

- Les enseignants peuvent créer et lister des cours
- Les enseignants peuvent voir leurs cours

Gestion des évaluations

- Les enseignants peuvent créer des évaluations simples
- Les étudiants peuvent voir les évaluations à venir

Notation

- Les enseignants peuvent attribuer des notes aux étudiants pour les évaluations

Autres fonctionnalités

- Les enseignants peuvent voir leurs cours
- Les enseignants peuvent voir les étudiants inscrits à leurs cours
- Les étudiants peuvent consulter leurs notes
- Les étudiants peuvent voir leurs cours
- Notification par e-mail lorsque des notes de cours sont attribuées

➤ **Contraintes et exigences techniques**

- ➔ POO avancée en PHP (héritage, abstraction, polymorphisme, encapsulation).
- ➔ Modélisation UML (diagramme de cas d'utilisation, diagramme de classes).
- ➔ Base de données relationnelle avec SQL (relations, clés étrangères, requêtes optimisées).
- ➔ Développement d'une API avec Laravel en suivant les bonnes pratiques REST.
- ➔ Interface utilisateur dynamique avec JavaScript (DOM, Ajax).
- ➔ Respect des bonnes pratiques (Clean Code, SOLID, MVC)