

Université Euro-Méditerranéenne de Fès

École d'Ingénierie Digitale et d'Intelligence Artificielle (EIDIA)

Filière : Cybersécurité

Rapport de Projet

GestioSeances

Application de Gestion des Demandes de
Changement et Annulation de Séances

Module : Développement Web (PHP/MySQL)

Année Universitaire : 2025-2026

Réalisé par :

Jaafar ouazzani chahdi

Iliass Zarquan

Aymane drissi bourhanbour

Aya Sefri

Encadré par :

Pr. Amamou Ahmad

Janvier 2026

Table des matières

1	Introduction	5
1.1	Contexte du Projet	5
1.2	Problématique	5
1.3	Objectifs du Projet	6
1.4	Organisation de l'Équipe	6
2	Analyse et Conception	7
2.1	Analyse des Besoins	7
2.1.1	Besoins Fonctionnels	7
2.1.2	Besoins Non Fonctionnels	8
2.2	Diagramme de Cas d'Utilisation	8
2.3	Diagramme de Classes	8
2.4	Modèle de Données	9
2.4.1	Schéma de la Base de Données	9
2.4.2	Énumérations	9
3	Réalisation	10
3.1	Architecture Technique	10
3.1.1	Architecture MVC	10
3.1.2	Structure du Projet	10
3.2	Technologies Utilisées	12
3.3	Implémentation des Fonctionnalités	12
3.3.1	Système d'Authentification	12
3.3.2	Gestion des Demandes	12
3.3.3	Système de Notifications	12
3.3.4	Module de Gestion des Utilisateurs (Admin)	13
3.3.5	Module Statistiques et Exports	13
3.4	Captures d'Écran	14
3.4.1	Interface Professeur	14
3.4.2	Interface Assistante	15
3.4.3	Interface Directeur	16

3.4.4	Gestion des Utilisateurs	16
3.4.5	Tableau de Bord Statistiques	17
4	Sécurité	18
4.1	Mesures de Sécurité Implémentées	18
4.1.1	Protection contre les Injections SQL	18
4.1.2	Protection XSS	18
4.1.3	Protection CSRF	18
4.1.4	Hachage des Mots de Passe	19
4.2	Contrôle d'Accès	19
5	Tests et Validation	21
5.1	Plan de Tests	21
5.2	Résultats des Tests	21
5.3	Données de Test - Demandes	22
5.4	Comptes de Test	22
6	Conclusion	23
6.1	Bilan du Projet	23
6.2	Compétences Acquisées	23
6.3	Fonctionnalités Implémentées	24
6.4	Perspectives d'Amélioration	24
A	Guide d'Installation	25
A.1	Prérequis	25
A.2	Étapes d'Installation	25
A.3	Configuration	25
A.4	URLs Principales	26

Table des figures

2.1	Description du diagramme de cas d'utilisation	8
3.1	Schéma de l'architecture MVC	10
3.2	Tableau de bord du professeur - Liste des demandes	14
3.3	File d'attente des demandes - Vue assistante	15
3.4	Demandes à approuver - Vue directeur	16
3.5	Interface de gestion des utilisateurs (Directeur)	16
3.6	Tableau de bord statistiques (Directeur)	17

Liste des tableaux

1.1	Répartition des tâches au sein de l'équipe	6
2.1	Classes principales du système	8
2.2	Structure de la base de données	9
2.3	Types énumérés utilisés	9
3.1	Stack technologique	12
4.1	Matrice des permissions	20
5.1	Synthèse des tests réalisés	21
5.2	Liste des demandes de test	22
5.3	Comptes utilisateurs de test	22
6.1	Récapitulatif des fonctionnalités	24
A.1	Routes de l'application	26

Chapitre 1

Introduction

1.1 Contexte du Projet

Dans le cadre de notre formation en Ingénierie Cybersécurité à l'EIDIA (École d'Ingénierie Digitale et d'Intelligence Artificielle) de l'Université Euro-Méditerranéenne de Fès, nous avons été amenés à réaliser un projet de développement web utilisant les technologies PHP et MySQL.

Ce projet s'inscrit dans une démarche pédagogique visant à mettre en pratique les concepts appris en cours, notamment :

- L'architecture MVC (Modèle-Vue-Contrôleur)
- La programmation orientée objet en PHP
- La gestion de bases de données relationnelles avec MySQL
- La sécurité des applications web
- Le travail collaboratif en équipe

1.2 Problématique

La gestion des demandes de changement ou d'annulation de séances universitaires représente un défi administratif majeur pour les établissements d'enseignement supérieur. Le processus traditionnel, souvent basé sur des formulaires papier ou des échanges d'emails, présente plusieurs inconvénients :

- Manque de traçabilité des demandes
- Délais de traitement importants
- Risque de perte d'informations
- Difficulté à gérer les conflits d'emploi du temps
- Absence de notifications automatiques

1.3 Objectifs du Projet

L'objectif principal de ce projet est de développer une application web permettant de centraliser et automatiser le processus de gestion des demandes de modification d'emploi du temps. Les objectifs spécifiques sont :

1. Permettre aux professeurs de soumettre des demandes de changement ou d'annulation de séances
2. Offrir aux assistantes administratives un outil de validation des demandes
3. Donner au directeur la possibilité d'approuver ou rejeter les demandes validées
4. Assurer la traçabilité complète du workflow de validation
5. Notifier automatiquement les parties prenantes à chaque étape

1.4 Organisation de l'Équipe

Le projet a été réalisé par une équipe de 4 développeurs, chacun ayant des responsabilités spécifiques :

TABLE 1.1 – Répartition des tâches au sein de l'équipe

Membre	Responsabilités
Jaafar	Architecture MVC, Core système, Configuration, Routing
Iliass	Modèles de données, Contrôleurs, Demandes et Notifications
Ayman	Vues et interfaces utilisateur, Layouts, Design
Aya	Documentation UML, Base de données, Tests, Guide utilisateur

Chapitre 2

Analyse et Conception

2.1 Analyse des Besoins

2.1.1 Besoins Fonctionnels

L'analyse des besoins a permis d'identifier les fonctionnalités suivantes :

Pour le Professeur :

- Se connecter/déconnecter de l'application
- Créer une demande de changement ou d'annulation
- Joindre des pièces justificatives
- Consulter l'état de ses demandes
- Annuler une demande en cours
- Recevoir des notifications

Pour l'Assistante Administrative :

- Consulter la file d'attente des demandes
- Vérifier les conflits d'emploi du temps
- Valider ou rejeter une demande
- Consulter le planning général
- Ajouter des commentaires

Pour le Directeur :

- Approuver ou rejeter définitivement une demande
- Gérer les utilisateurs du système
- Consulter les statistiques
- Paramétrer l'application

2.1.2 Besoins Non Fonctionnels

- **Sécurité** : Protection contre les injections SQL, XSS, CSRF
- **Performance** : Temps de réponse inférieur à 2 secondes
- **Ergonomie** : Interface intuitive et responsive
- **Disponibilité** : Application accessible 24h/24
- **Maintenabilité** : Code structuré et documenté

2.2 Diagramme de Cas d'Utilisation

Le diagramme de cas d'utilisation présente les interactions entre les acteurs et le système.

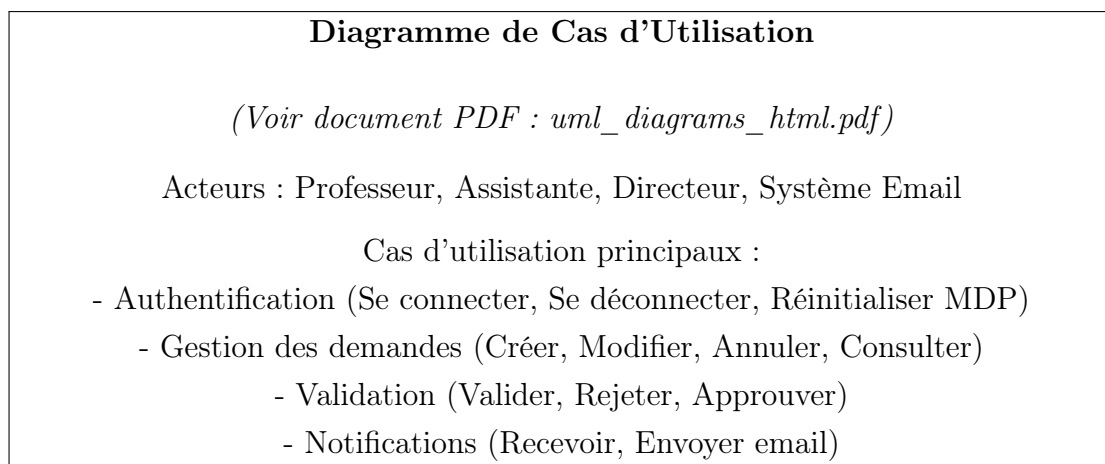


FIGURE 2.1 – Description du diagramme de cas d'utilisation

2.3 Diagramme de Classes

Le diagramme de classes modélise la structure statique du système.

TABLE 2.1 – Classes principales du système

Classe	Description
User	Gestion des utilisateurs (professeurs, assistantes, directeur)
Demande	Représente une demande de changement/annulation
Validation	Enregistre les validations effectuées
Notification	Gère les notifications aux utilisateurs
PieceJointe	Gère les fichiers justificatifs uploadés
Seance	Représente une séance de cours
Salle	Gère les salles de classe
Matiere	Représente les matières enseignées

2.4 Modèle de Données

2.4.1 Schéma de la Base de Données

La base de données `gestioseances` comprend 8 tables principales :

TABLE 2.2 – Structure de la base de données

Table	Clé Primaire	Description
users	id	Utilisateurs du système
matieres	id	Matières enseignées
salles	id	Salles de cours
seances	id	Séances programmées
demandes	id	Demandes de changement/annulation
validations	id	Historique des validations
pieces_jointes	id	Fichiers justificatifs
notifications	id	Notifications utilisateurs

2.4.2 Énumérations

TABLE 2.3 – Types énumérés utilisés

Énumération	Valeurs
RoleUser	professeur, assistante, directeur
StatutDemande	brouillon, en_attente, validee_assistante, approuvee, rejetee, annulee
TypeDemande	changement, annulation
TypeSeance	cours, td, tp, examen

Chapitre 3

Réalisation

3.1 Architecture Technique

3.1.1 Architecture MVC

L'application suit le patron de conception MVC (Modèle-Vue-Contrôleur) :

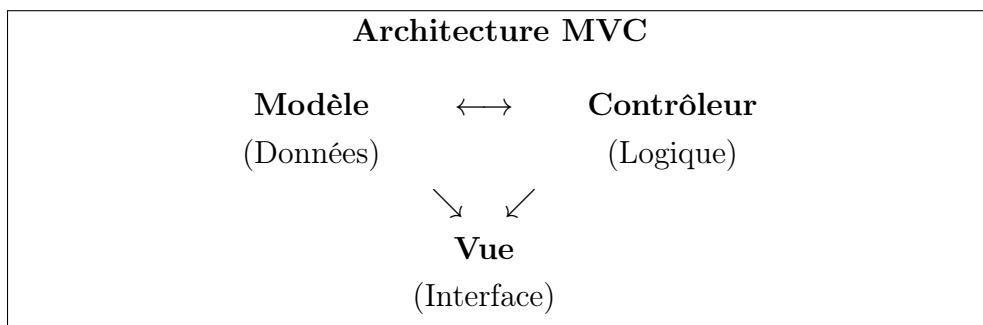


FIGURE 3.1 – Schéma de l'architecture MVC

3.1.2 Structure du Projet

Listing 3.1 – Arborescence du projet

```
1 GestioSeances/  
2 |-- app/  
3 |   |-- controllers/  
4 |       |-- AdminController.php  
5 |       |-- AuthController.php  
6 |       |-- Controller.php  
7 |       |-- DemandeController.php  
8 |       |-- NotificationController.php  
9 |       +-- StatsController.php  
10 |   |-- core/
```

```
11 | | |-- Database.php
12 | | |-- Router.php
13 | | |-- Security.php
14 | | +-- Session.php
15 | |-- models/
16 | | |-- Demande.php
17 | | |-- Model.php
18 | | |-- Notification.php
19 | | |-- PieceJointe.php
20 | | |-- User.php
21 | | +-- Validation.php
22 | +-- views/
23 |     |-- admin/
24 |         |-- users.php
25 |         |-- user-form.php
26 |         +-- settings.php
27 |     |-- auth/
28 |     |-- demandes/
29 |     |-- layouts/
30 |     |-- notifications/
31 |     +-- stats/
32 |         +-- index.php
33 |-- database/
34 | +-- schema_final.sql
35 |-- public/
36 | |-- index.php
37 | +-- .htaccess
38 |-- storage/
39 | +-- uploads/
40 +-- config.php
```

3.2 Technologies Utilisées

TABLE 3.1 – Stack technologique

Catégorie	Technologie	Version
Backend	PHP	8.1+
Base de données	MySQL	8.0
Serveur Web	Apache	2.4
Frontend	HTML5, CSS3	-
Framework CSS	Bootstrap	5.3
Icônes	Bootstrap Icons	1.11
Graphiques	Chart.js	4.0
Environnement	XAMPP	8.2

3.3 Implémentation des Fonctionnalités

3.3.1 Système d'Authentification

Le système d'authentification implémente :

- Hachage des mots de passe avec `bcrypt`
- Protection contre le brute force (blocage après 5 tentatives)
- Tokens CSRF pour les formulaires
- Sessions sécurisées avec régénération d'ID

3.3.2 Gestion des Demandes

Le workflow de gestion des demandes suit le processus :

1. **Création** : Le professeur crée une demande (statut : brouillon)
2. **Soumission** : Le professeur soumet la demande (statut : en_attente)
3. **Validation** : L'assistante valide ou rejette (statut : validee_assistante ou rejetee)
4. **Approbation** : Le directeur approuve ou rejette (statut : approuvee ou rejetee)

3.3.3 Système de Notifications

Les notifications sont générées automatiquement lors de chaque changement de statut et sont visibles dans l'interface utilisateur avec un badge indiquant le nombre de notifications non lues.

3.3.4 Module de Gestion des Utilisateurs (Admin)

Le module d'administration permet au directeur de gérer les utilisateurs du système :

- **Liste des utilisateurs** : Affichage de tous les utilisateurs avec leur rôle, email, téléphone et statut
- **Création** : Ajout de nouveaux utilisateurs (professeurs, assistantes, directeurs)
- **Modification** : Édition des informations et changement de mot de passe
- **Désactivation** : Désactivation des comptes (sans suppression définitive)

Listing 3.2 – Extrait du contrôleur AdminController.php

```
1 public function users(): void
2 {
3     $this->requireRole(ROLE_DIRECTEUR);
4     $users = $this->getAllUsers();
5     $this->view('admin/users', [
6         'users' => $users,
7         'flash' => $this->getFlash()
8     ]);
9 }
10
11 public function createUser(): void
12 {
13     $this->requireRole(ROLE_DIRECTEUR);
14     // Validation et cr ation de l'utilisateur
15     $hashedPassword = Security::hashPassword($password);
16     // INSERT INTO users...
17 }
```

3.3.5 Module Statistiques et Exports

Le tableau de bord statistiques offre une vue d'ensemble des demandes :

- **Cartes de statistiques** : Total, En attente, Approuvées, Rejetées
- **Graphique d'évolution** : Demandes sur les 6 derniers mois (Chart.js)
- **Répartition par type** : Graphique donut (Changement vs Annulation)
- **Top professeurs** : Classement par nombre de demandes
- **Activité récente** : Fil des dernières actions
- **Export CSV** : Téléchargement des données pour Excel
- **Export PDF** : Rapport imprimable

Listing 3.3 – Méthode d'export CSV

```

1 public function exportExcel(): void
2 {
3     $this->requireRole(ROLE_DIRECTEUR);
4     $demandes = $this->getAllDemandes();
5
6     header('Content-Type: text/csv; charset=utf-8');
7     header('Content-Disposition: attachment; filename="demandes.csv"');
8
9     $output = fopen('php://output', 'w');
10    fprintf($output, chr(0xEF).chr(0xBB).chr(0xBF)); // BOM UTF-8
11
12    fputcsv($output, ['ID', 'Professeur', 'Type', ...], ';');
13    foreach ($demandes as $d) {
14        fputcsv($output, [...], ';');
15    }
16 }

```

3.4 Captures d'Écran

3.4.1 Interface Professeur

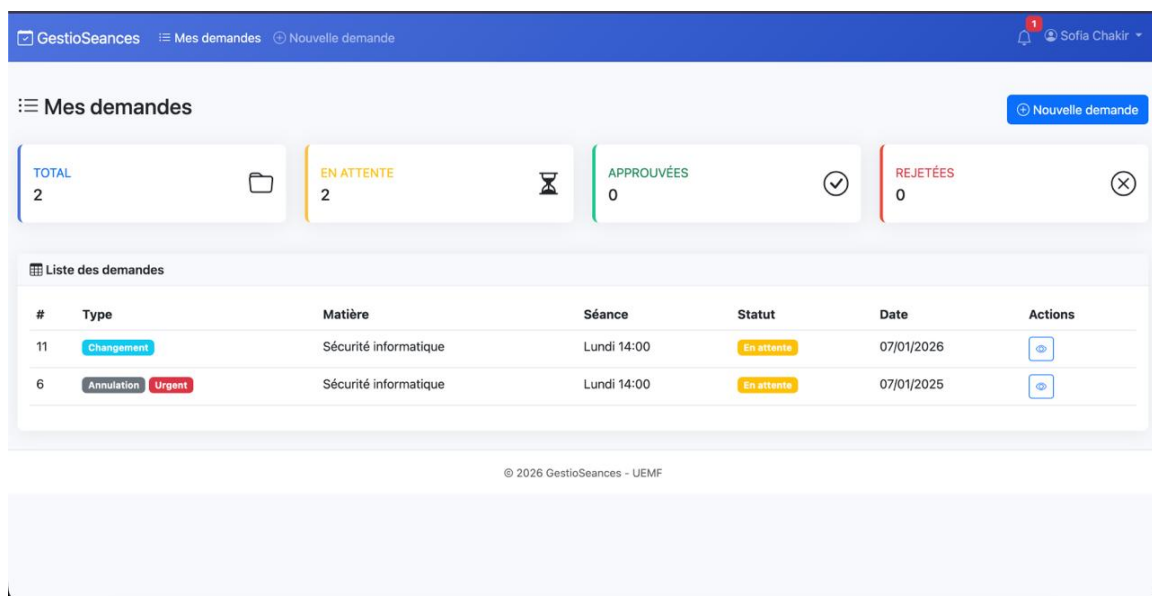


FIGURE 3.2 – Tableau de bord du professeur - Liste des demandes

L'interface professeur présente :

- Un tableau de bord avec les statistiques (Total, En attente, Approuvées, Rejetées)
- La liste des demandes avec leur type, matière, séance, statut et date
- Des badges colorés pour identifier rapidement le type et le statut
- Un bouton pour créer une nouvelle demande

3.4.2 Interface Assistante

GestioSeances - File d'attente						
File d'attente des demandes						
⚠ Demandes urgentes (3)						
#	Professeur	Type	Matière	Date	Actions	
3	Karim Ziani	Changement	Génie Logiciel	05/01/2025	⚙ Traiter	
5	Ahmed Benjelloun	Changement	Intelligence Artificielle	06/01/2025	⚙ Traiter	
6	Sofia Chakir	Annulation	Sécurité informatique	07/01/2025	⚙ Traiter	
☰ Demandes en attente de validation (3)						
#	Professeur	Type	Matière	Séance	Soumise le	Actions
6	Sofia Chakir	Annulation	Sécurité informatique	Lundi 14:00	07/01/2025 07:00	⚙ Examiner
7	Mohammed Tazi	Changement	Programmation Web	Vendredi 08:30	07/01/2025 09:30	⚙ Examiner
11	Sofia Chakir	Changement	Sécurité informatique	Lundi 14:00	07/01/2026 19:03	⚙ Examiner

FIGURE 3.3 – File d’attente des demandes - Vue assistante

L’interface assistante affiche :

- Les demandes urgentes en priorité (section rouge)
- Les demandes en attente de validation
- Les informations détaillées : professeur, type, matière, date
- Des boutons d’action pour traiter ou examiner chaque demande

3.4.3 Interface Directeur

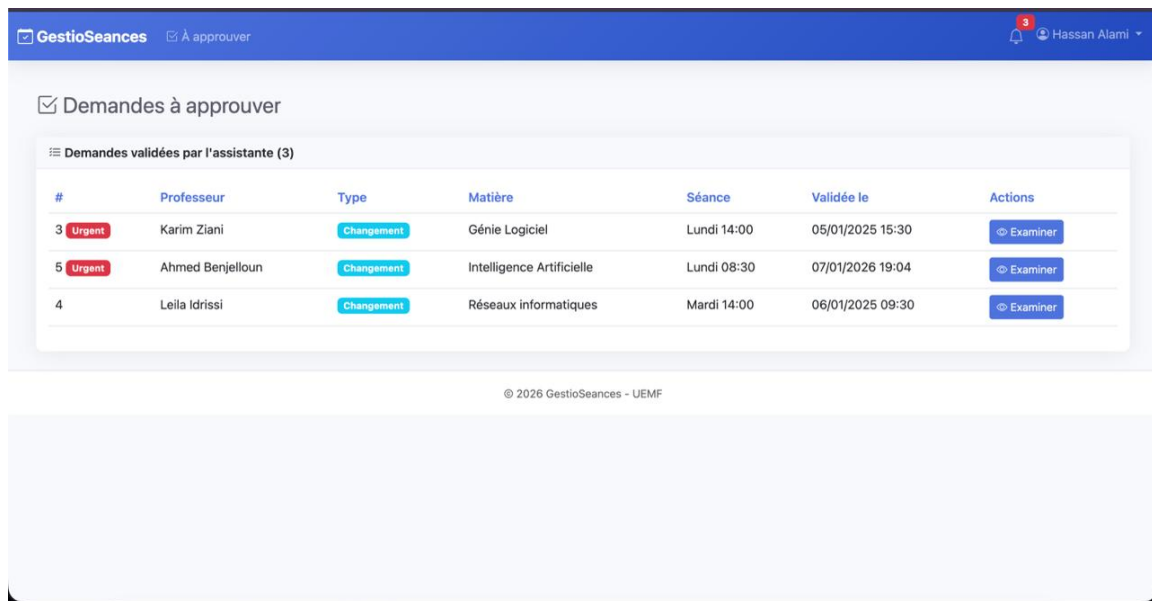


FIGURE 3.4 – Demandes à approuver - Vue directeur

L'interface directeur présente :

- Les demandes validées par l'assistante
- Les badges urgents pour les demandes prioritaires
- La date de validation par l'assistante
- Des boutons pour examiner et prendre une décision finale

3.4.4 Gestion des Utilisateurs

Interface de Gestion des Utilisateurs					
#	Nom	Email	Rôle	Téléphone	Statut
1	Hassan Alami	hassan.alami@eidia.ma	Directeur	0661234567	Actif
2	Fatima Bennani	fatima.bennani@eidia.ma	Assistante	0662345678	Actif
3	Mohammed Tazi	mohammed.tazi@eidia.ma	Professeur	0663456789	Actif
4	Sofia Chakir	sofia.chakir@eidia.ma	Professeur	0664567890	Actif
5	Karim Ziani	karim.ziani@eidia.ma	Professeur	0665678901	Actif
6	Nadia El Amrani	nadia.amrani@eidia.ma	Professeur	0666789012	Actif
Boutons : Nouvel Utilisateur / Modifier / Désactiver					

FIGURE 3.5 – Interface de gestion des utilisateurs (Directeur)

3.4.5 Tableau de Bord Statistiques

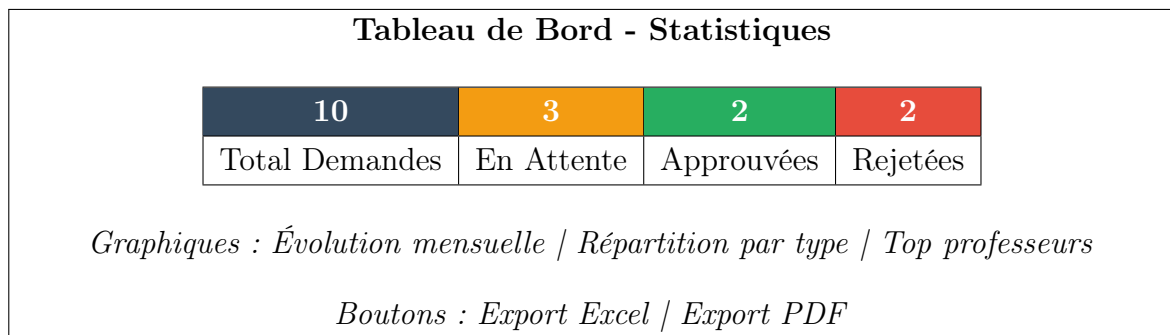


FIGURE 3.6 – Tableau de bord statistiques (Directeur)

Chapitre 4

Sécurité

4.1 Mesures de Sécurité Implémentées

4.1.1 Protection contre les Injections SQL

Toutes les requêtes SQL utilisent des requêtes préparées avec PDO :

Listing 4.1 – Exemple de requête préparée

```
1 public function findByEmail(string $email): ?array
2 {
3     $sql = "SELECT * FROM users WHERE email = :email";
4     $stmt = $this->db->prepare($sql);
5     $stmt->execute(['email' => $email]);
6     return $stmt->fetch() ?: null;
7 }
```

4.1.2 Protection XSS

Toutes les données affichées sont échappées avec `htmlspecialchars()` :

Listing 4.2 – Échappement XSS

```
1 public static function e(?string $string): string
2 {
3     return htmlspecialchars($string ?? '', ENT_QUOTES, 'UTF-8');
4 }
```

4.1.3 Protection CSRF

Les formulaires incluent un token CSRF vérifié côté serveur :

Listing 4.3 – Génération et vérification CSRF

```
1 public static function generateCsrfToken(): string
2 {
3     $token = bin2hex(random_bytes(32));
4     $_SESSION[CSRF_TOKEN_NAME] = $token;
5     return $token;
6 }
7
8 public static function verifyCsrfToken(?string $token): bool
9 {
10     return isset($_SESSION[CSRF_TOKEN_NAME])
11         && hash_equals($_SESSION[CSRF_TOKEN_NAME], $token ?? '');
12 }
```

4.1.4 Hachage des Mots de Passe

Les mots de passe sont hachés avec bcrypt :

Listing 4.4 – Hachage bcrypt

```
1 public static function hashPassword(string $password): string
2 {
3     return password_hash($password, PASSWORD_BCRYPT);
4 }
5
6 public static function verifyPassword(string $password, string
7     $hash): bool
8 {
9     return password_verify($password, $hash);
10 }
```

4.2 Contrôle d'Accès

Le système implémente un contrôle d'accès basé sur les rôles (RBAC) :

TABLE 4.1 – Matrice des permissions

Action	Professeur	Assistante	Directeur
Créer une demande	✓	✗	✗
Valider une demande	✗	✓	✗
Approuver une demande	✗	✗	✓
Gérer les utilisateurs	✗	✗	✓
Consulter les statistiques	✗	✗	✓
Exporter les données	✗	✗	✓
Consulter le planning	✓	✓	✓

Chapitre 5

Tests et Validation

5.1 Plan de Tests

Les tests ont été réalisés selon les catégories suivantes :

- **Tests fonctionnels** : Vérification de chaque fonctionnalité
- **Tests de sécurité** : Tentatives d'injection, XSS, CSRF
- **Tests d'interface** : Responsive design, compatibilité navigateur
- **Tests de performance** : Temps de réponse, charge

5.2 Résultats des Tests

TABLE 5.1 – Synthèse des tests réalisés

Module	Tests	Passés	Taux
Authentification	5	5	100%
Gestion des demandes	8	8	100%
Validation (Assistante)	4	4	100%
Approbation (Directeur)	3	3	100%
Notifications	3	3	100%
Gestion utilisateurs	4	4	100%
Statistiques et exports	3	3	100%
Sécurité	10	10	100%
Total	40	40	100%

5.3 Données de Test - Demandes

TABLE 5.2 – Liste des demandes de test

ID	Professeur	Type	Matière	Statut	Urgent
1	Tazi Mohammed	changement	Programmation Web	approuvee	Non
2	El Amrani Nadia	annulation	Base de données	approuvee	Non
3	Ziani Karim	changement	Génie Logiciel	validee_assist.	Oui
4	Idrissi Leila	changement	Réseaux inform.	validee_assist.	Non
5	Benjelloun Ahmed	changement	Intelligence Artif.	en_attente	Oui
6	Chakir Sofia	annulation	Sécurité inform.	en_attente	Oui
7	Tazi Mohammed	changement	Programmation Web	en_attente	Non
8	El Amrani Nadia	changement	Base de données	rejetee	Non
9	Ziani Karim	annulation	Génie Logiciel	rejetee	Non
10	Idrissi Leila	changement	Réseaux inform.	brouillon	Non

5.4 Comptes de Test

TABLE 5.3 – Comptes utilisateurs de test

Rôle	Email	Mot de passe
Directeur	hassan.alami@eidia.ma	password123
Assistante	fatima.bennani@eidia.ma	password123
Assistante	samira.mansouri@eidia.ma	password123
Professeur	mohammed.tazi@eidia.ma	password123
Professeur	sofia.chakir@eidia.ma	password123
Professeur	karim.ziani@eidia.ma	password123

Chapitre 6

Conclusion

6.1 Bilan du Projet

Le projet GestioSeances a permis de développer une application web complète et fonctionnelle répondant aux besoins identifiés. Les objectifs principaux ont été atteints :

- ✓ Architecture MVC propre et maintenable
- ✓ Système d'authentification sécurisé
- ✓ Workflow complet de gestion des demandes
- ✓ Interface utilisateur intuitive et responsive
- ✓ Système de notifications fonctionnel
- ✓ Module de gestion des utilisateurs (CRUD)
- ✓ Tableau de bord statistiques avec graphiques
- ✓ Exports de données (CSV/PDF)
- ✓ Documentation complète (UML, guide utilisateur, tests)

6.2 Compétences Acquises

Ce projet nous a permis de développer plusieurs compétences :

- Maîtrise du développement PHP orienté objet
- Conception et implémentation d'une architecture MVC
- Gestion de bases de données relationnelles
- Implémentation de mesures de sécurité web
- Création de tableaux de bord avec Chart.js
- Génération d'exports de données (CSV, PDF)
- Travail collaboratif et gestion de projet
- Rédaction de documentation technique

6.3 Fonctionnalités Implémentées

TABLE 6.1 – Récapitulatif des fonctionnalités

Fonctionnalité	Statut	Remarque
Authentification	✓	Bcrypt, CSRF, blocage brute force
Gestion des demandes	✓	CRUD complet avec workflow
Validation assistante	✓	File d'attente, priorité urgences
Approbation directeur	✓	Décision finale
Notifications	✓	Badge temps réel
Gestion utilisateurs	✓	CRUD, activation/désactivation
Statistiques	✓	4 indicateurs, 3 graphiques
Export Excel (CSV)	✓	UTF-8, compatible Excel
Export PDF	✓	Rapport imprimable
Paramètres système	✓	Configuration application

6.4 Perspectives d'Amélioration

Plusieurs améliorations pourraient être apportées dans le futur :

- Intégration d'un calendrier interactif (FullCalendar)
- Envoi d'emails automatiques (PHPMailer)
- Application mobile (PWA)
- Statistiques avancées avec filtres par période
- Intégration avec le système d'information de l'université
- API REST pour intégrations tierces

Annexe A

Guide d'Installation

A.1 Prérequis

- XAMPP 8.2 ou supérieur (PHP 8.1+, MySQL 8.0, Apache 2.4)
- Navigateur web moderne (Chrome, Firefox, Safari, Edge)

A.2 Étapes d'Installation

1. Extraire le dossier `Gestioseances` dans `C:\xampp\htdocs\`
2. Démarrer Apache et MySQL depuis le panneau de contrôle XAMPP
3. Ouvrir phpMyAdmin : `http://localhost/phpmyadmin`
4. Créer une base de données nommée `gestioseances`
5. Importer le fichier `database/schema_final.sql`
6. Accéder à l'application : `http://localhost/Gestioseances/public`
7. Se connecter avec un compte de test

A.3 Configuration

Le fichier `config.php` contient les paramètres de configuration :

Listing A.1 – Configuration de la base de données

```
1 define('DB_HOST', 'localhost');  
2 define('DB_NAME', 'gestioseances');  
3 define('DB_USER', 'root');  
4 define('DB_PASS', '');  
5 define('APP_URL', 'http://localhost/Gestioseances/public');
```

A.4 URLs Principales

TABLE A.1 – Routes de l'application

URL	Description
/login	Page de connexion
/demandes	Liste des demandes (Professeur)
/demandes/file-attente	File d'attente (Assistante)
/demandes/a-approuver	Demandes à approuver (Directeur)
/admin/users	Gestion des utilisateurs (Directeur)
/stats	Tableau de bord statistiques (Directeur)
/stats/export/excel	Export CSV
/stats/export/pdf	Export PDF
/admin/settings	Paramètres (Directeur)