

# Rapport de projet : Développement d'une plateforme web de gestion des séminaires de l'IMSP

## 1. Introduction

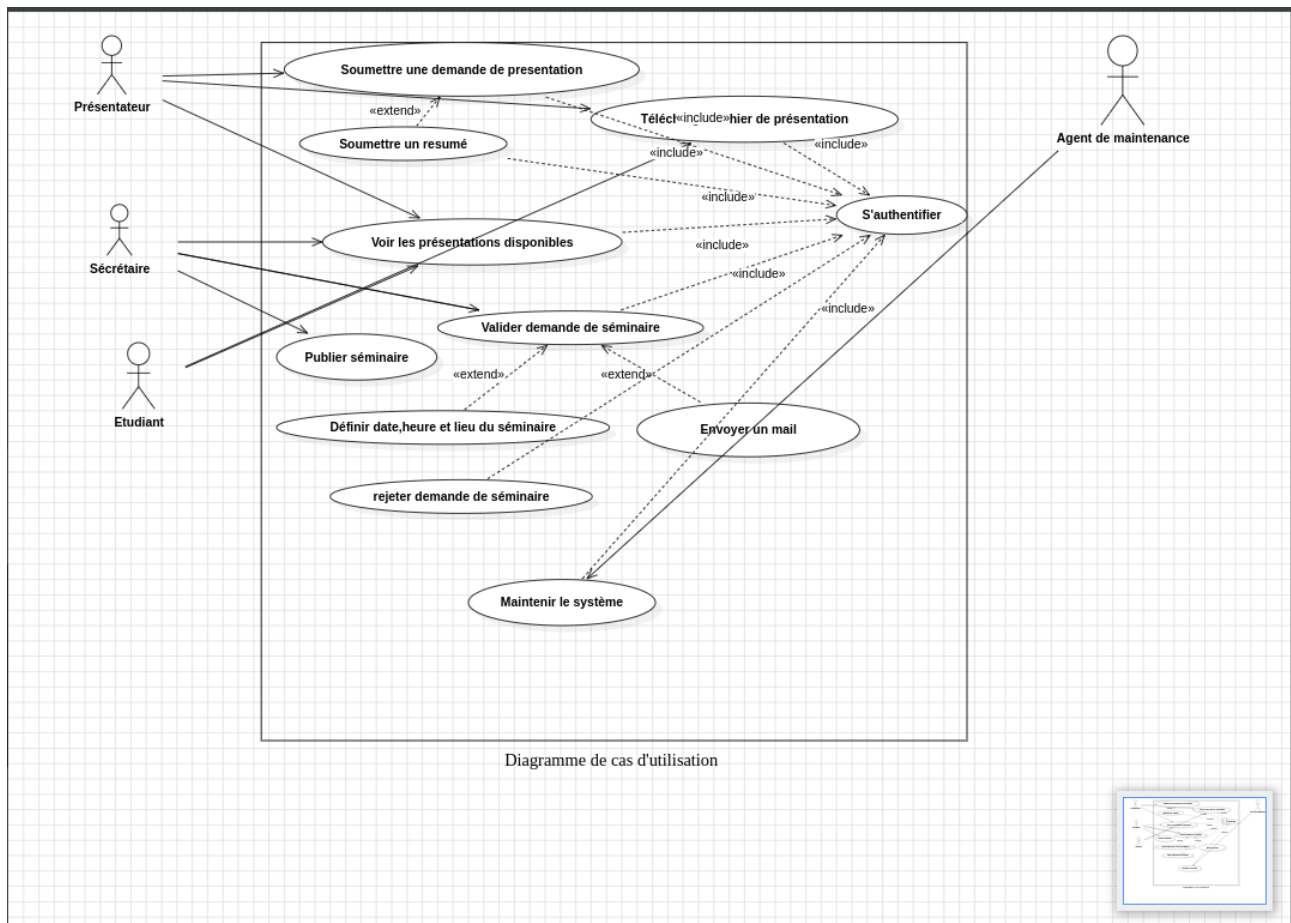
L'IMSP organise des séminaires généraux hebdomadaires (tous les mercredis matin à Dangbo) où des chercheurs présentent leurs travaux à la communauté scientifique de l'Institut. La gestion manuelle de ces événements (inscription des présentateurs, planification du lieu et de l'horaire, diffusion des documents) est chronophage et complexe. En effet, l'organisation de séminaires est « une activité cruciale [...] souvent chronophage, exigeant une coordination complexe et une attention méticuleuse aux détails ». Dans ce contexte, les études sur la gestion de conférences soulignent l'intérêt de solutions informatiques : les logiciels de gestion de conférences sont conçus comme « une plateforme numérique qui automatise et centralise la planification, l'exécution et le suivi » des événements. De même, les manuels sur la gestion d'événements rappellent que ce processus couvre l'ensemble des tâches nécessaires, de la planification à la communication (gestion des participants, envoi de mails, etc.)

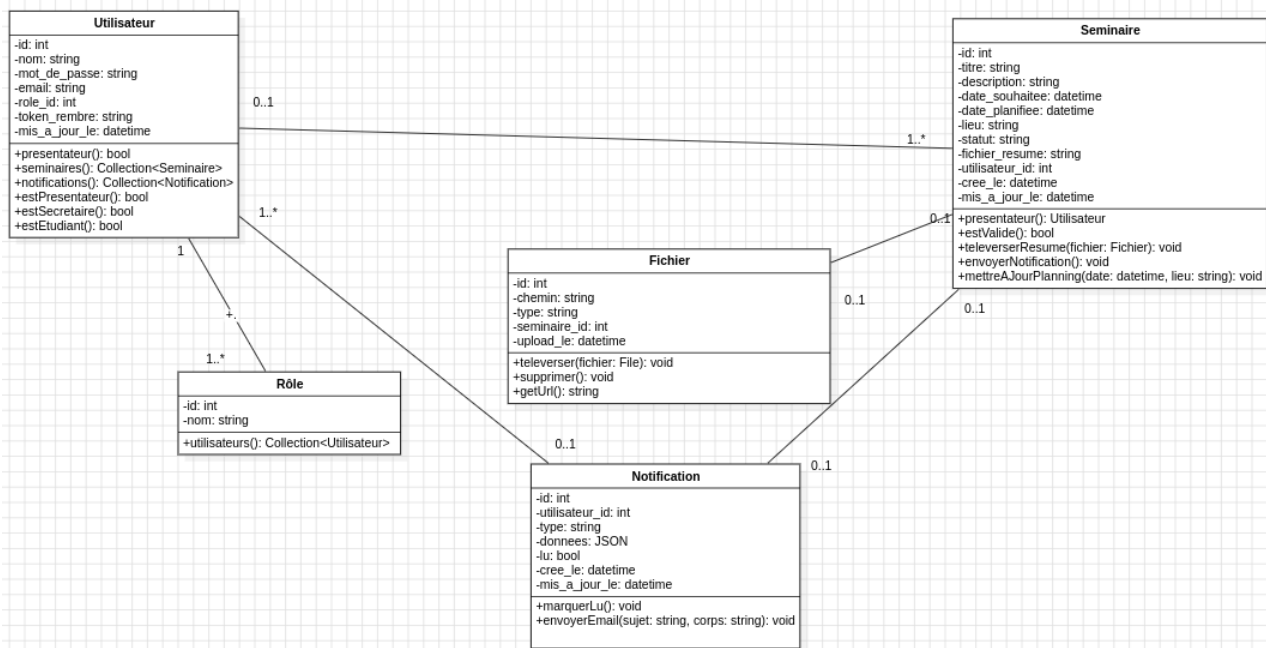
## 2. Cahier des charges fonctionnel

Le cahier des charges fonctionnel définit précisément les besoins du futur système en termes de fonctionnalités 5 . Pour notre projet, les spécifications essentielles sont les suivantes :

- **Authentification sécurisée avec gestion des rôles.** Les utilisateurs doivent se connecter avec un compte protégé. Trois rôles sont prévus : « présentateur », « secrétaire scientifique » et « étudiant », chacun ayant des droits particuliers.
- **Soumission de demandes de séminaires.** Le rôle « présentateur » accède à un formulaire où il indique le thème, la date souhaitée (optionnelle à ce stade), le titre et une description du séminaire à présenter. Ce formulaire utilise un mécanisme CSRF pour la sécurité.
- **Validation ou rejet par le secrétaire.** Les demandes soumises sont affichées dans un tableau réservé au secrétaire. Celui-ci peut examiner chaque demande et choisir de la valider ou la rejeter. Les demandes rejetées restent accessibles dans un historique, tandis que les demandes validées passent à l'étape suivante.
- **Planification (date, heure, lieu).** Lorsque le secrétaire valide une demande, il spécifie la date, l'heure et le lieu du séminaire. Cette information s'inscrit dans le planning global. Le secrétaire peut modifier ultérieurement ces détails si besoin.
- **Notifications par e-mail.** Après validation d'un séminaire, un courriel automatique est envoyé au présentateur pour l'informer de la confirmation et des détails (date, lieu) du séminaire. Le système utilisera le système de notifications de Laravel pour effectuer cet envoi 6 .

- **Téléversement du résumé.** Dix jours avant la date du séminaire, le présentateur doit soumettre en ligne un résumé de sa présentation (généralement un document PDF). Le site enregistre ce fichier pour le mettre à disposition.
- **Affichage des séminaires programmés.** Les étudiants (utilisateurs en lecture seule) peuvent consulter la liste des séminaires validés, avec leurs thèmes, intervenants, dates et lieux. Cette liste doit être accessible via une interface publique (après authentification ou ouverte) pour informer la communauté.
- **Téléchargement des fichiers de présentation.** Après chaque séminaire, les fichiers de présentation (slides, documents) sont téléversés et associés à l'événement. Les étudiants peuvent ensuite les télécharger depuis l'application.
- **Recherche et tri.** L'interface étudiante propose une barre de recherche et des filtres dynamiques pour retrouver un séminaire par thème, date, intervenant ou statut (validé/en attente). Cette recherche doit être rapide et interactive.





Ces fonctionnalités reprennent les besoins exprimés et permettent une gestion complète du cycle des séminaires. Notons que, comme le rappelle la documentation de projet, le cahier des charges fonctionnel « permet de définir les besoins auxquels le futur produit devra répondre, en termes de fonctionnalités ». Nous y avons donc listé tous les services attendus pour répondre précisément aux besoins de l'IMSP.

### 3. Architecture technique du projet

L'application est bâtie sur le framework PHP Laravel, qui impose une architecture MVC (Model-View-Controller). Laravel offre ainsi une séparation claire entre la logique métier (modèles), le traitement (contrôleurs) et la présentation (vues). Les données sont gérées via **Eloquent ORM**, le système d'ActiveRecord de Laravel. Comme l'explique la documentation, « Eloquent ORM [...] fournit une implémentation ActiveRecord simple et élégante pour travailler avec votre base de données : chaque table a un modèle correspondant utilisé pour interagir avec cette table ». Par exemple, nous avons créé un modèle Seminaire (table seminaires ) et un modèle User (table users ) avec leurs relations (un présentateur peut avoir plusieurs séminaires, etc.).

La couche de données repose sur MySQL. La connexion à la base est configurée via le fichier config/database.php , qui exploite les variables d'environnement (dans .env ) pour préciser le hôte, le nom de la base, l'utilisateur et le mot de passe. Ce choix de MySQL est justifié par sa popularité dans le monde web : Oracle souligne que MySQL « sous-tend un vaste éventail de sites web et d'applications »

et qu'il offre de hautes performances et fiabilité, capable de gérer de gros volumes de données et de nombreuses connexions simultanées. La documentation Laravel confirme également que la configuration par défaut (MySQL) est prête à l'emploi via les variables d'environnement, simplifiant ainsi le développement local et la mise en production.

Le front-end utilise Blade, le moteur de templates de Laravel, pour générer dynamiquement les pages HTML. Blade compile les vues en PHP natif (fichiers `.blade.php`), ce qui « ajoute essentiellement zéro surcharge » de performance. Les vues Blade reçoivent des données passées par les contrôleurs et utilisent des directives (`@csrf`, `@foreach`, etc.) pour construire les formulaires et listes d'éléments. Pour le design et le style, nous avons adopté Tailwind CSS, un framework « utility-first » qui propose de nombreuses classes utilitaires prêtes à l'emploi (par ex. `flex`, `pt-4`, `text-center`) permettant de construire rapidement des interfaces responsives directement dans le HTML. Comme le souligne la documentation officielle, Laravel avec Blade s'intègre bien à Tailwind, simplifiant la mise en page et la maintenance du CSS.

La sécurité de l'application repose sur les composants standards de Laravel. Le framework propose un système d'authentification complet, basé sur des « guards » et « providers », pour gérer les connexions. Nous avons utilisé un starter kit Laravel (par exemple Laravel Breeze ou Jetstream) qui a généré automatiquement les contrôleurs, routes et vues de l'authentification <sup>15</sup>. Pour l'autorisation, Laravel offre deux mécanismes principaux : les gates et les policies <sup>16</sup>. Les gates sont utiles pour contrôler des actions globales (ex. accéder au tableau de bord), tandis que les policies ciblent les actions liées à un modèle particulier (par ex. modifier son propre séminaire). Nous avons ainsi assigné des permissions selon les rôles : par exemple, seul un utilisateur avec le rôle « secrétaire » peut valider une demande, tandis qu'un « présentateur » ne peut accéder qu'à ses propres saisines. Cette structure de contrôle d'accès multi-rôles s'appuie sur les middleware fournis par Laravel et/ou le package Spatie (si utilisé) pour vérifier en temps réel les permissions (ex. `auth` et `can:viewSeminaire`).

Enfin, l'application inclut un système de notifications et d'envoi d'e-mails. Laravel propose un composant Notifications complet qui peut diffuser des messages par plusieurs canaux (email, SMS, Slack, etc.). Dans notre cas, nous utilisons le canal e-mail pour notifier les présentateurs. Après la validation d'une demande, un objet Notification est envoyé, ce qui génère l'e-mail au format texte/HTML. L'application peut aussi stocker les notifications en base pour un affichage ultérieur dans l'interface (non exploité ici). Cette architecture (Laravel + Blade + Tailwind + MySQL) assure un projet robuste et maintenable, conforme aux meilleures pratiques du développement web actuel.

#### **4. Description des interfaces et des fonctionnalités par rôle**

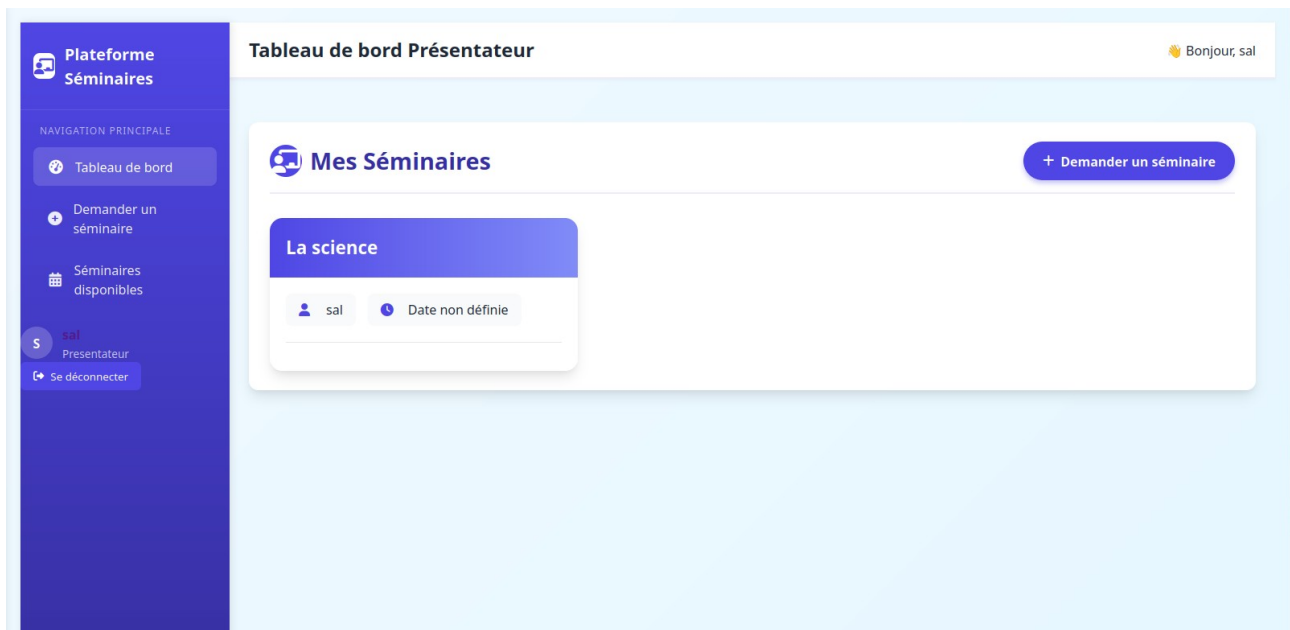
L'application propose trois interfaces principales, adaptées aux trois rôles définis : présentateur, secrétaire scientifique et étudiant. Chaque interface affiche les éléments pertinents pour l'utilisateur connecté et applique les autorisations correspondantes.

- **Présentateur**

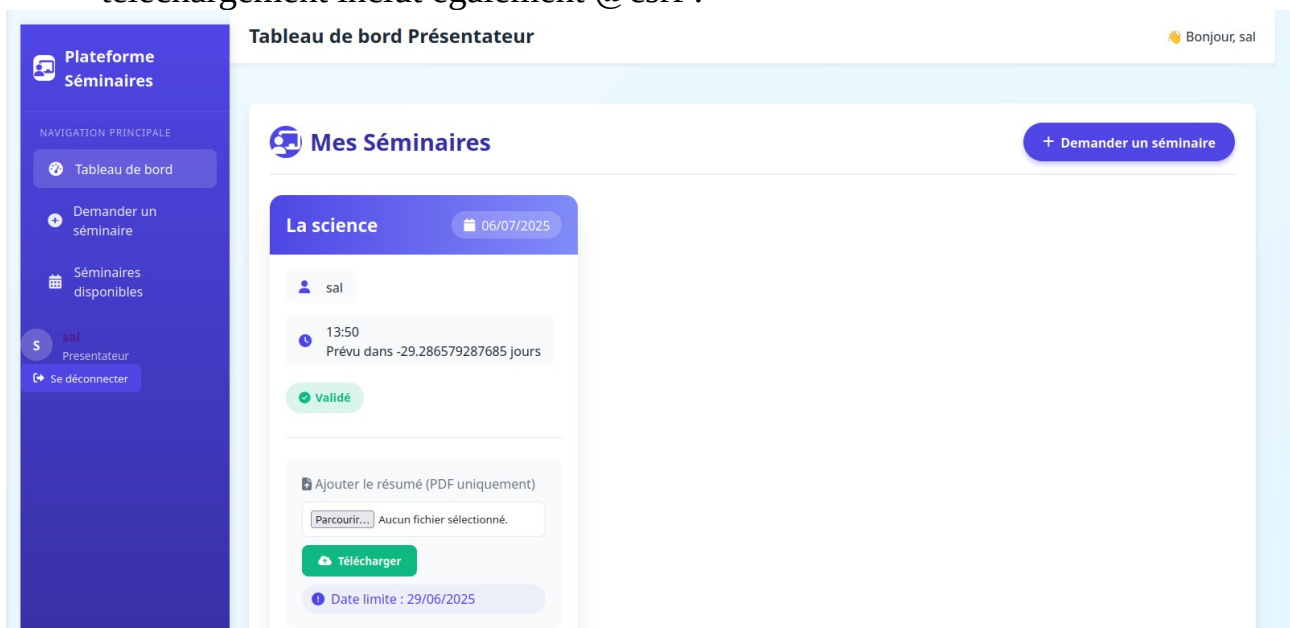
- **Soumission d'une demande.** Un présentateur peut remplir un formulaire pour proposer un nouveau séminaire. Ce formulaire, réalisé en Blade, comprend les champs titre, thème, description, et éventuellement la date souhaitée. Il inclut systématiquement un champ caché CSRF (via la directive `@csrf`) pour sécuriser la requête POST. En envoyant ce formulaire, le système crée une nouvelle entrée en base avec le statut « en attente ».

The screenshot shows the 'Plateforme Séminaires' web application. On the left is a dark blue sidebar with navigation links: 'Tableau de bord', 'Demander un séminaire' (active), 'Séminaires disponibles', and a user profile section for 'Présentateur' with a 'Se déconnecter' link. The main content area is titled 'Demander un séminaire' and features a light blue header with a 'Bonjour, sal' greeting. The form itself is titled 'Demander un nouveau séminaire' and includes a section for 'Informations importantes' with instructions and bullet points. Below this are input fields for 'Thème du séminaire' (containing 'La science'), 'Résumé (optionnel)' (containing 'La science...'), and 'Date souhaitée (optionnel)' (containing '06 / 07 / 2025'). A blue button 'Envoyer la demande' is at the bottom, with a 'Retour au tableau de bord' link below it.

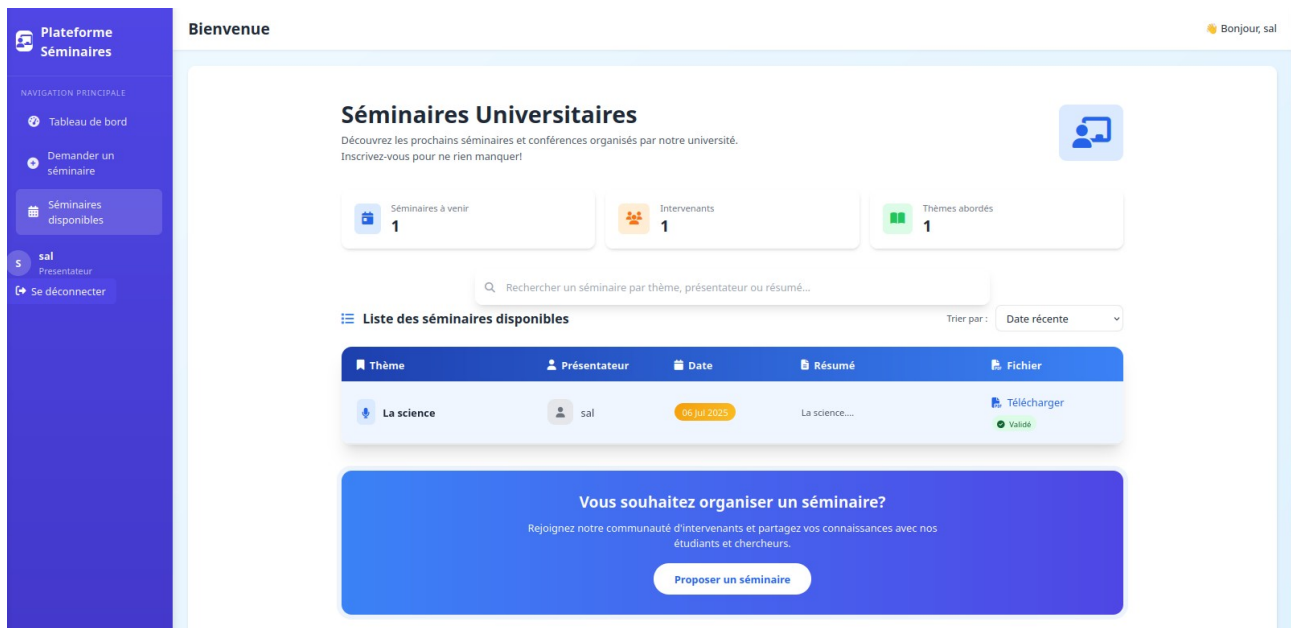
- **Suivi de ses demandes.** Une page dédiée liste tous les séminaires soumis par ce présentateur, avec leur statut (en attente, validé, rejeté). Le présentateur peut ainsi voir rapidement l'évolution de ses demandes. Une fois une demande validée par le secrétaire, le présentateur reçoit une notification (par e-mail) lui indiquant la date et le lieu retenus, grâce au système de notifications de Laravel. Cette notification automatique est gérée par la fonction d'envoi de notification mail intégrée.



- **Téléversement du résumé.** Dans son espace, le présentateur voit la date du séminaire validé et un bouton pour téléverser le résumé. Le système lui rappelle de soumettre ce résumé au moins 10 jours avant la présentation. Le fichier uploadé (par exemple, résumé en PDF) est enregistré sur le serveur (fonction `$request->file('fichier')->store()`) via le composant Filesystem de Laravel, qui facilite le stockage de fichiers uploadés. Le formulaire de téléchargement inclut également `@csrf`.

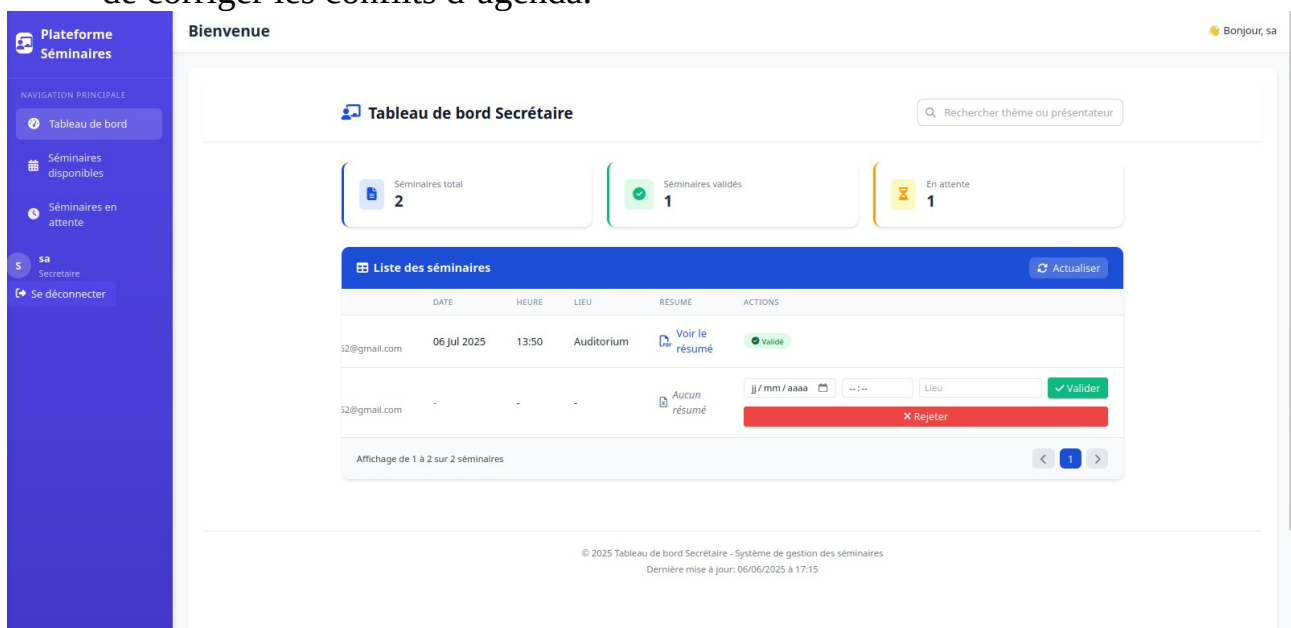


- **Visualisation des séminaires.** Le présentateur dispose enfin d'un aperçu de ses séminaires validés, avec accès au résumé une fois déposé et aux éventuels fichiers de présentation téléchargés après l'événement.



## • Secrétaire scientifique

- **Validation des demandes.** L'interface du secrétaire affiche un tableau récapitulatif de toutes les demandes de séminaires en attente. Pour chaque ligne, le secrétaire dispose d'options pour valider ou rejeter la demande. Cette opération active ou désactive le séminaire. Lors de la validation, une fenêtre/modale s'ouvre (ou un formulaire) permettant de saisir la date, l'heure et le lieu du séminaire. Ces informations sont sauvegardées en base. Si le séminaire est rejeté, il reste archivé en statut « rejeté » pour historique, mais n'apparaît pas dans les prochains plannings.
- **Planification (date/heure/lieu).** Le secrétaire peut à tout moment modifier le planning d'un séminaire validé en accédant à un formulaire d'édition (dates, lieu). Les changements sont mis à jour dans la base. Cette flexibilité lui permet de corriger les conflits d'agenda.



- **Publication des séminaires.** Une fois validé et planifié, le séminaire passe au statut « publié ». Il devient alors visible dans l'interface étudiante. Le secrétaire

peut également ajouter (ou demander aux présentateurs d'ajouter) les fichiers de présentation, qu'il peut téléverser pour les associer au séminaire concerné.

- **Gestion des notifications.** Après validation, le système envoie automatiquement un e-mail au présentateur (via Laravel Notification) pour l'informer, sans intervention manuelle du secrétaire.

En cas de changement de planning, un nouvel e-mail peut aussi être généré.

- **Étudiant**

- **Consultation de la liste des séminaires.** L'étudiant connecté (ou l'utilisateur invité) accède à une page publique listant les séminaires programmés. Le tableau présente pour chaque événement le thème, le titre, le nom du présentateur, la date et le lieu. Seuls les séminaires validés sont affichés.

- **Recherche dynamique.** Un champ de recherche en haut de la liste permet de filtrer instantanément les séminaires par mot-clé (thème, intervenant, etc.). Cette recherche est implémentée en JavaScript (par exemple via AJAX+jQuery) : à chaque frappe, une requête est envoyée au serveur et les résultats filtrés s'affichent sans recharger la page. Comme le note un tutoriel sur Laravel, une telle « live search » améliore grandement l'expérience utilisateur en fournissant des résultats en temps réel, ce qui est particulièrement utile avec de nombreux événements. Les étudiants peuvent donc rapidement trouver les séminaires qui les intéressent.

- **Téléchargement des présentations.** Une fois le séminaire terminé, l'étudiant voit apparaître un lien de téléchargement sur la ligne correspondante pour obtenir le fichier de présentation (slides, etc.) que le présentateur a fourni. Le clic déclenche un téléchargement direct du fichier stocké sur le serveur. Cette fonctionnalité repose sur les routes protégées de Laravel et renvoie le fichier en réponse HTTP.

The screenshot shows a web application for 'Séminaires Universitaires'. The sidebar on the left has a logo and navigation links: 'Séminaires disponibles', 'sall Etudiant', and 'Se déconnecter'. The main content area has a header 'Bienvenue' and a greeting 'Bonjour, sall'. Below this is a section titled 'Séminaires Universitaires' with a description and a search bar. A table lists seminars with columns: 'Thème', 'Présentateur', 'Date', 'Résumé', and 'Fichier'. The table shows one seminar: 'La science' by 'sall' on '06 juil 2025'. Below the table is a blue box with the text 'Vous souhaitez organiser un séminaire?' and a button 'Proposer un séminaire'.

Thème	Présentateur	Date	Résumé	Fichier
La science	sall	06 juil 2025	La science...	Télécharger Validé

<http://localhost:8000/storage/seminaires/118C0q08xnXX0ZfOCabdebQ7ltdU45AlHd7sx1Q.pdf>

Dans chacune de ces interfaces, les contrôles d'accès veillent à ce qu'un utilisateur ne voie que ce qui le concerne : par exemple, le présentateur ne peut pas approuver de



demande (fonction réservée au secrétaire) et l'étudiant ne dispose d'aucun formulaire d'édition. La documentation Laravel sur les formulaires recommande d'inclure un jeton CSRF dans chaque formulaire HTML, pratique que nous avons systématiquement suivie pour sécuriser les requêtes des présentateurs et du secrétaire.

### Sources :

- Documentation Laravel (authentification, Blade, Eloquent, notifications); sites et articles thématiques sur la gestion d'événements et de conférences officiel de l'IMSP;
- Séminaires IMSP → <https://www.imsp-benin.com/home/seminaires.php>

- Organisation de séminaires - Gain de productivité grâce à l'IA  
→ <https://www.demarretonaventure.com/gain-productivite-ia/gain-productivite-ia-organisation-de-seminaires/>

- We Found the 10 Best Conference Management Software Solutions  
→ <https://www.eventsair.com/blog/conference-management-software>

- Les étapes incontournables pour une gestion d'événements réussie – Blog  
→ <https://imagina.com/fr/blog/article/gestion-evenement/>

- Comment faire un cahier des charges ? Définition, étapes et exemples  
<https://www.appvizer.fr/magazine/operations/gestion-de-projet/cahier-des-charges>

- Notifications - Laravel 12.x - The PHP Framework For Web Artisans  
<https://laravel.com/docs/12.x/notifications>

- Streamline Your Development with Laravel and Tailwind CSS | by Asis Sharma | Medium  
<https://medium.com/@asis-sharma/streamline-your-development-with-laravel-and-tailwind-css-74f8168f6545>

- Eloquent ORM - Laravel 5.0 - The PHP Framework For Web Artisans  
<https://laravel.com/docs/5.0/eloquent>

- Database: Getting Started - Laravel 12.x - The PHP Framework For Web Artisans  
<https://laravel.com/docs/12.x/database>

- MySQL: Understanding What It Is and How It's Used  
<https://www.oracle.com/mysql/what-is-mysql/>

- Blade Templates - Laravel 12.x - The PHP Framework For Web Artisans  
<https://laravel.com/docs/12.x/blade>

-Tailwind CSS - Rapidly build modern websites without ever leaving your HTML.  
<https://tailwindcss.com/>

-Authentication - Laravel 12.x - The PHP Framework For Web Artisans  
<https://laravel.com/docs/12.x/authentication>

-Authorization - Laravel 12.x - The PHP Framework For Web Artisans  
<https://laravel.com/docs/12.x/authorization>

-File Storage - Laravel 11.x - The PHP Framework For Web Artisans  
<https://laravel.com/docs/11.x/filesystem>

-Laravel 10 Search Box with Live Results Using AJAX jQuery  
<https://devrims.com/blog/live-search-laravel-ajax/>