

I. Description : Gestion d'une conférence scientifique

Dans le cadre de cet atelier, nous allons développer une application web de **gestion de conférences scientifiques**. L'objectif est de simuler le processus complet d'organisation et de participation à une conférence académique en intégrant plusieurs fonctionnalités essentielles.

L'étude de cas repose sur le diagramme de classe suivant :

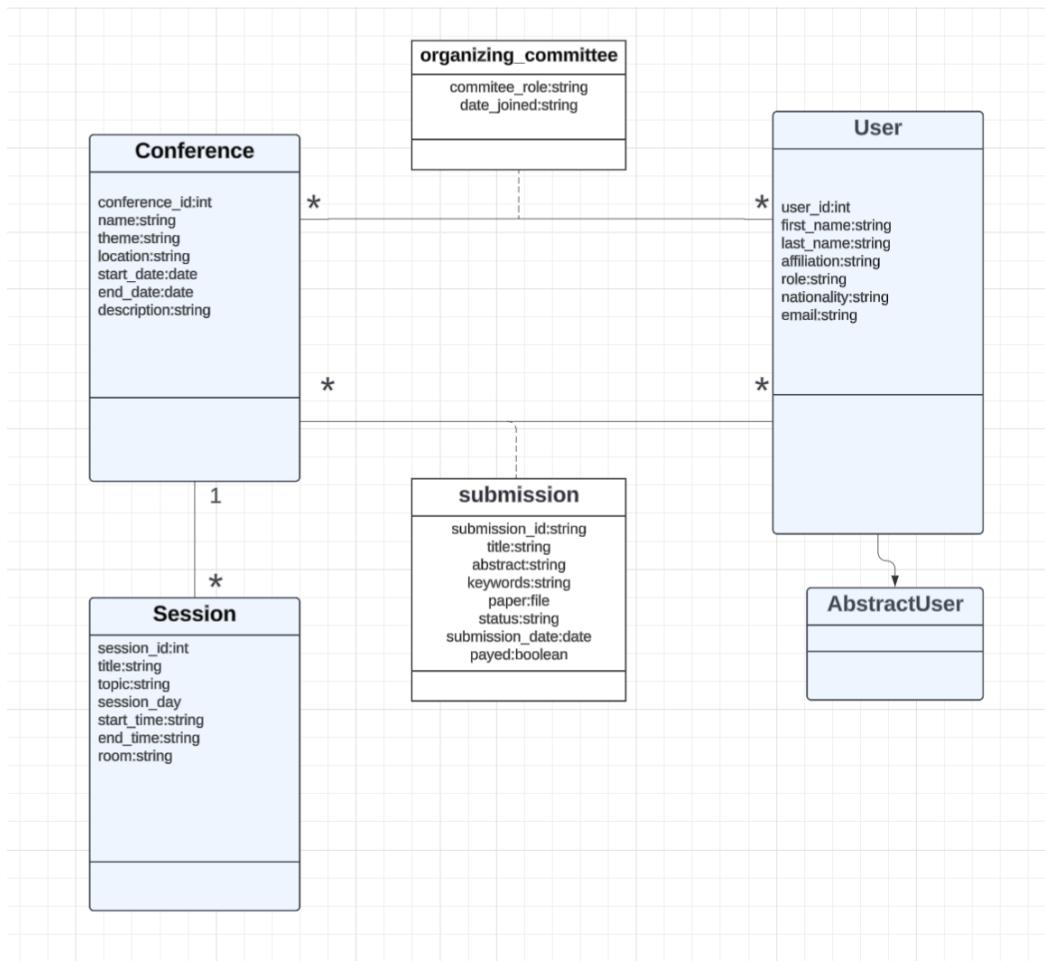


Figure1 : diagramme de classe

V. Partie Client (Views) [Create Retrieve Update Delete] (CRUD)

7. Gestion Utilisateurs

- a. Implémenter une fonctionnalité permettant aux utilisateurs de :
 - Créer un compte comme étant des participants. Après la création du compte, il sera redirigé vers une page de login.
 - Le champ role n'apparaît pas dans le formulaire (l'utilisateur ne peut pas le choisir).
 - Lors de la création, le champ rôle est automatiquement défini à "**participant**".
 - S'authentifier avec un username et un mot de passe.
- b. Ajouter une barre de navigation comme le montre les figures suivantes :
 - Si l'utilisateur n'est pas connecté
 - Si l'utilisateur est connecté
- c. Limiter l'accès aux participants qui ne sont pas de type comité organisation de manipuler les conférences.
- d. Donner des autorisations à seulement les utilisateurs connectés d'accéder l'interface de création, modification et suppression des conférences en utilisant **LoginRequiredMixin**.

8. Affichage de la liste des soumissions – **ListSubmissions**

- a. Créer une vue générique de type **ListView** pour afficher toutes les soumissions associées à une conférence pour l'utilisateur connecté.
- b. Afficher dans la liste les informations principales :
 - Titre de la soumission
 - Statut (Submitted, Under Review, Accepted, Rejected)
 - Utilisateur (auteur)
 - Conférence associée
 - Date de soumission
 - Paiement (payé ou non)
 - Ajouter un bouton "Détails" pour chaque soumission.

9. Détail d'une soumission – **DetailSubmission**

- a. Créer une vue générique de type **DetailView** pour afficher les informations complètes d'une soumission.
- b. Lorsqu'un utilisateur clique sur "**Détails**" dans la liste, la vue doit afficher :
 - Titre, résumé (abstract), mots-clés, fichier PDF, statut, conférence liée, auteur.
- c. Ajouter un bouton pour télécharger l'article PDF.

10. Ajout d'une soumission – **AddSubmission**

- a. Créer une vue générique **CreateView** pour permettre à un utilisateur d'ajouter une soumission.
- b. Rediriger l'utilisateur vers la liste des soumissions après ajout.

11. Modification d'une soumission – **UpdateSubmission**

- a. Créer une vue générique **UpdateView** pour modifier une soumission déjà existante.
(Une soumission avec l'état accepté ou rejeté ne peut pas être modifiable)
- b. Les champs modifiables sont : title, abstract, keywords, paper.
 - c. Les champs non modifiables : submission_id, submission_date, user.
 - d. Rediriger l'utilisateur vers la liste des soumissions après modification.