**A. Classes Principales :**

* **Utilisateur**
* Attributs :
* id: int
* nom : string
* email : string
* motPasse : string
* membre : string (valeurs possibles : createur, gerant, regulier)
* Méthodes :
* inscription() : void

permet à un nouvel utilisateur de créer un compte.

* seConnecter() : void

permet à un utilisateur de se connecter à son compte.

* **Groupe**
* Attributs :
* id : int
* nom: string
* description: string
* idCreateur : Utilisateur // référence à Utilisateur
* membres : MembreGroupe // liste des membres ===ArrayList<MembreGroupe>
* Méthodes :
* AjouterMembre(membre : Utilisateur) : void // référence à Utilisateur

permet l’ ajout d’un nouveau membre au groupe

* supprimerMembre(membre : Utilisateur) : void // référence à Utilisateur

permet le retour de la liste des chambres disponibles

* programmerActivite(activite : Activite) : void // liste d’ activités

permet de programmer une nouvelle activité pour le groupe.

* listerMembres(utilisateur : Utilisateur) : String // liste d’utilisateur

retourne la liste des membres du groupe

* **MembreGroupe**
* Attributs :
* groupeId : Groupe // référence à Groupe
* utilisateurId : Utilisateur // référence à Utilisateur
* role : string (valeurs possibles : gerant, regulier)
* Méthodes :
* activer () : void

Active le statut d'un membre

* desactiver () : void

Désactive le statut d'un membre

* **Activite**
* Attributs :
* id : int
* titre : string
* description : string
* date : Date
* groupeId : Groupe // référence à Groupe
* participants : Utilisateur // référence à Utilisateur ===//ArrayList<Utilisateur> // liste des Utilisateurs participant à l’activité
* Méthodes :
* confirmerParticipation(utilisateur : Utilisateur) : void

permet à un utilisateur de confirmer sa participation à l'activité.

* infirmerParticipation(utilisateur : Utilisateur) : void

Permet à un utilisateur d'infirmer sa participation à l'activité.

* listerParticipants(utilisateur : Utilisateur) : String // liste des participants

Retourne la liste des participants à l'activité.

* **Notification**
* Attributs :
* id : int
* message : string
* date : Date
* utilisateurId : Utilisateur // référence à Utilisateur
* Méthodes :
* envoyer() : void

Envoie une notification à l'utilisateur.

* lire() : void

Marque la notification comme lue.

* **Commentaire**
* Attributs :
* id : int
* utilisateurId : Utilisateur // référence à Utilisateur
* activiteId : Activite // référence à Activite
* texte : string
* date : Date
* Méthodes :
* ajouterCommentaire() : void

Ajoute un commentaire à une activité.

* supprimerCommentaire() : void
* Supprime un commentaire d'une activité.

**B. Relations entre classes :**

**Résumé des relations :**

* **UN utilisateur fait partie de 0,1 ou plusieurs groupe(association)**
* **Un groupe doit avoir(a) minimum un membre et sans les membres, il n’y a pas de groupe**
* **Un groupe doit avoir(a) minimum une et sans les activités, il n’y a pas d’activité, un groupe ne peut être créé.**
* Un utilisateur fait zéro ou plusieurs commentaires
* Une nodification est envoyé a un ou plusieurs utilisateurs

Diagramme de classes :

