







les **acteurs** (Employé, Manager, RH) et le système.

- **Raison** : Ce diagramme aide à définir clairement **les fonctionnalités principales** (comme "Demander un congé" ou "Consulter les congés") en montrant qui fait quoi. Cela facilite la communication avec les parties prenantes non techniques.

## 2. Diagramme de classes

- **Objectif** : Définir la **structure du système** en modélisant les entités (classes) et leurs relations.
  - **Raison** : Ce diagramme est utile pour créer la base de données ou l'architecture du code. Il formalise les **attributs**, **relations** et rôles, comme la hiérarchie entre **Utilisateur**, **Employé**, **Manager** et leurs interactions avec des objets comme **Congé**.
- 

### 3. Diagramme de séquence

- **Objectif** : Détailler les **flux d'interactions dynamiques** dans un scénario précis.
  - **Raison** : Le diagramme de séquence met en lumière les étapes d'un processus clé (ici, **la demande de congé**) en montrant l'ordre des actions entre les acteurs et le système. Cela est utile pour l'implémentation des fonctionnalités et pour les tests.
- 

### Pourquoi ces trois types ?

- **Complémentarité** : Ces diagrammes offrent une vue **complète** du système :
  - Le **cas d'utilisation** montre les besoins fonctionnels.
  - Le **diagramme de classes** décrit la structure interne.
  - Le **diagramme de séquence** explique le fonctionnement dynamique.
- **Équilibre** : Ces trois diagrammes suffisent pour un projet de cette taille, sans alourdir inutilement la documentation.