

1. Analyse du besoin

L'analyse du besoin est une phase cruciale dans le développement d'un logiciel. Elle permet de comprendre les attentes des utilisateurs et de définir les fonctionnalités principales du système.

Objectifs du projet :

- Concevoir un logiciel permettant à un manager, aux RH et aux employés de gérer les congés de manière fluide et sécurisée.
- S'assurer que chaque acteur a accès uniquement aux informations et fonctionnalités qui lui sont nécessaires.

Rôles et fonctionnalités :

- **Employé :**
 - Soumettre des demandes de congé via une interface.
 - Consulter l'état de ses demandes de congé.
- **Manager (N+1) :**
 - Approuver ou refuser les demandes de congé de ses subordonnés.
 - Consulter l'historique des congés de ses collaborateurs.
- **Ressources humaines (RH) :**
 - Consulter tous les congés de tous les employés.
 - Valider certaines demandes de congé.

Types de congés à gérer :

- **Vacances**
- **RTT**
- **Congé maladie**
- **Jours fériés**
- **Congés de droit** (mariage, naissance, etc.)

Réflexion sur la sécurité et la confidentialité :

- Les managers ne doivent voir que les congés de leurs subordonnés.
- La RH doit pouvoir consulter les congés de tous les employés, mais les employés ne doivent pas accéder aux congés des autres.
-

2. Modélisation avec UML

L’UML (Unified Modeling Language) est un outil clé de l'ingénierie logicielle pour la conception du système. Il permet de représenter les structures, les processus, et les interactions.

Diagramme de cas d’utilisation :

Voici un diagramme de cas d’utilisation UML détaillé avec les acteurs et les cas d’utilisation :

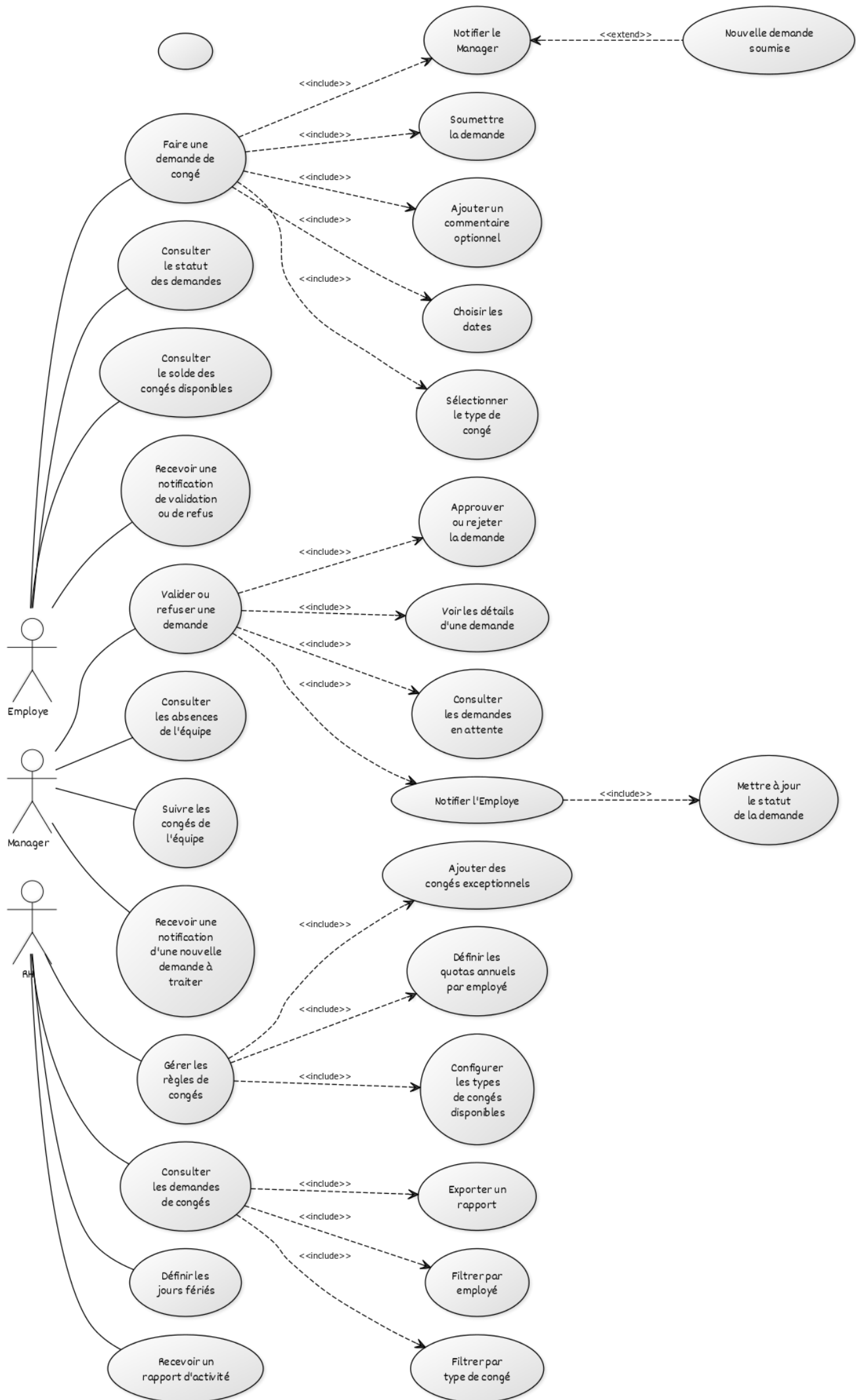
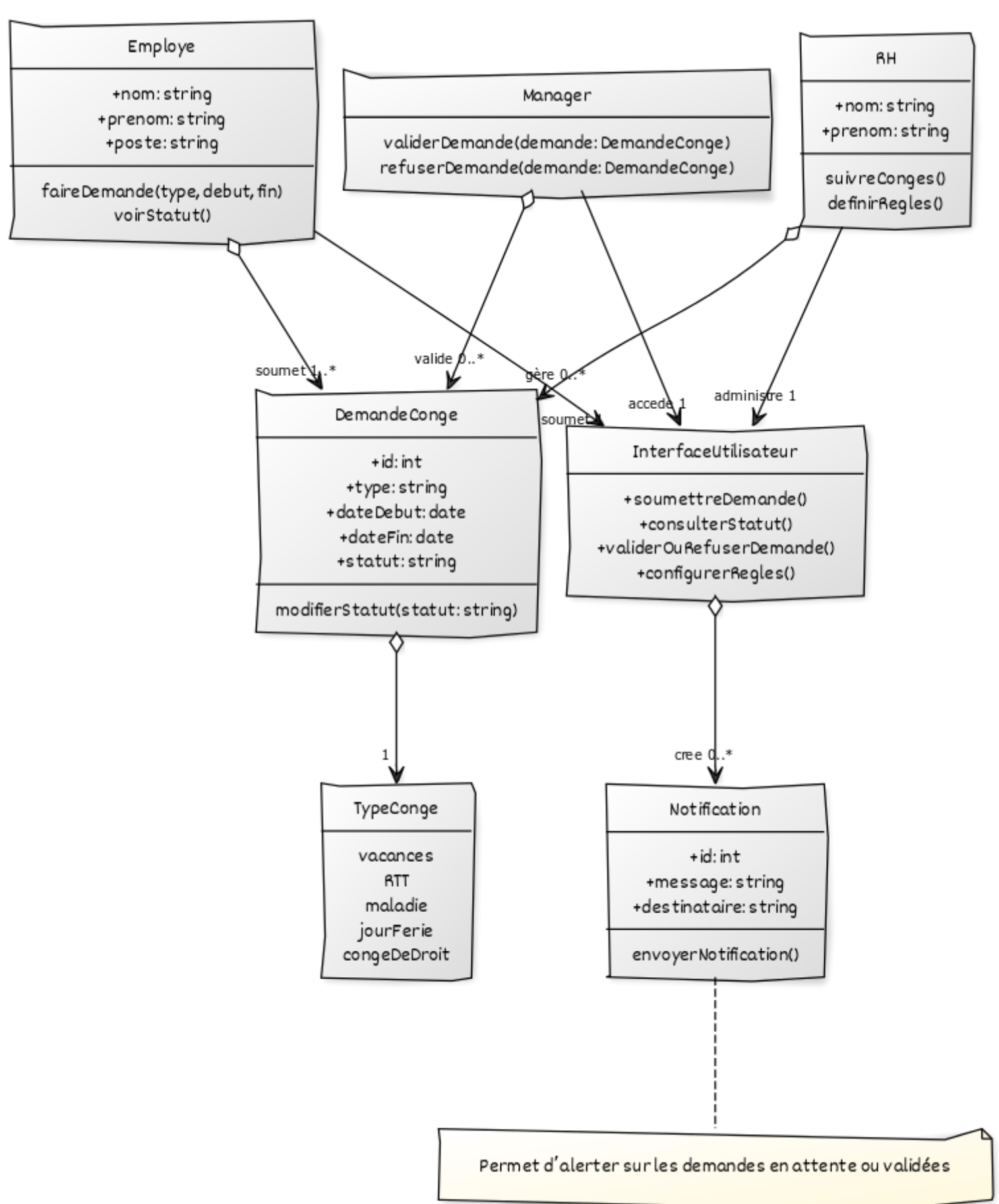


Diagramme de classes UML :

Le diagramme de classes est crucial pour représenter les entités principales du système et leurs relations.



Explication des classes et relations :

1. Classe Employé :

- **Attributs :**

- id: Identifiant unique de l'employé.
- nom: Nom de l'employé.
- prénom: Prénom de l'employé.
- email: Adresse e-mail de l'employé.
- soldeCongé: Nombre de jours de congé restants pour l'employé.
- faireDemande(): L'employé peut soumettre une demande de congé.

- **Méthodes :**

2. Classe Manager :

- **Attributs :**

- id: Identifiant unique du manager.
- nom: Nom du manager.
- email: Adresse e-mail du manager.

- **Méthodes :**

- validerDemande(): Le manager valide les demandes de congé de ses subordonnés.

2. Classe RH (Ressources Humaines) :

- **Attributs :**

- id: Identifiant unique de l'utilisateur RH.
- nom: Nom de l'utilisateur RH.
- email: Adresse e-mail de l'utilisateur RH.

- **Méthodes :**

- consulterCongé(): Permet au responsable RH de consulter l'état des congés de tous les employés.
- validerCongé(): Le responsable RH peut valider des congés une fois approuvés par le manager.

3. Classe TypeDeCongé :

- **Attributs :**

- type: Type de congé (par exemple, vacances, RTT, congé maladie, etc.).
- description: Description du type de congé (par exemple, "Congé pour mariage", "Congé maladie", etc.).

4. Classe Congé :

- **Attributs :**

- id: Identifiant unique du congé.
- type: Le type de congé (lié à la classe TypeDeCongé).
- dateDebut: La date de début du congé.
- dateFin: La date de fin du congé.
- statut: Statut du congé (par exemple, "En attente", "Validé", "Refusé").

- **Méthodes :**

- valider(): Permet de valider un congé.
- refuser(): Permet de refuser un congé.

5. Classe DemandeDeCongé :

- **Attributs :**

- id: Identifiant unique de la demande de congé.
- dateDemande: Date de la demande de congé.
- raison: Raison de la demande de congé.
- statut: Statut de la demande de congé (par exemple, "En attente", "Approuvée", "Refusée"). ○

- Méthodes :**

- soumettre(): Soumettre la demande de congé pour qu'elle soit traitée par le manager et/ou le RH.

Relations entre les classes :

- **Relation entre Employé et DemandeDeCongé** : Un employé peut faire plusieurs demandes de congé. C'est une relation 1:N entre les classes Employé et DemandeDeCongé.
- **Relation entre Manager et DemandeDeCongé** : Un manager peut valider plusieurs demandes de congé de ses employés. Cette relation peut être décrite comme 1:N entre Manager et DemandeDeCongé.
- **Relation entre Manager et Congé** : Le manager est responsable de valider ou refuser les congés, mais les congés sont validés au niveau de la demande dans DemandeDeCongé.
- **Relation entre RH et Congé** : Le RH peut consulter et valider des congés une fois que les managers les ont approuvés.
- **Relation entre TypeDeCongé et Congé** : Un congé a un type spécifique, ce qui implique une relation 1:N entre TypeDeCongé et Congé.

3. Établir un plan de test

Le plan de test doit couvrir toutes les fonctionnalités critiques du système et garantir qu'il répond bien aux exigences spécifiées.

Tests fonctionnels :

1. **Soumettre une demande de congé :**

- **Test** : Un employé soumet une demande via l'interface.
- **Critère d'acceptation** : La demande est correctement enregistrée dans la base de données, et une notification est envoyée au manager.
- **Test automatique** : Tester la soumission via un formulaire d'interface utilisateur.

2. **Validation de la demande par le manager :**

- **Test** : Le manager peut valider ou refuser une demande de congé.
- **Critère d'acceptation** : Le statut de la demande change en fonction de l'action du manager.
- **Test automatique** : Simuler un manager qui approuve ou rejette une demande.

3. **Consultation des congés par la RH :**

- **Test** : La RH peut consulter tous les congés des employés.
- **Critère d'acceptation** : La RH doit pouvoir consulter un tableau de bord avec les demandes de congé en temps réel.
- **Test automatique** : Tester la visibilité des demandes de congé dans l'interface RH.

Tests de sécurité :

- **Objectif** : Vérifier que la sécurité des données est bien respectée.
- **Test** : Assurez-vous que les managers ne peuvent voir que les congés de leurs subordonnés.
- **Critère d'acceptation** : Les rôles d'accès doivent être correctement définis et respecter la confidentialité des données.

Tests de performance :

- **Objectif** : Tester la performance sous une charge élevée de demandes.
 - **Test** : Soumettre 1000 demandes de congé simultanément et mesurer le temps de réponse.
 - **Critère d'acceptation** : Le système doit répondre en moins de 2 secondes pour une requête.
- #### 4. Implémentation des principales fonctionnalités

Les fonctionnalités à implémenter incluent :

- **Gestion des utilisateurs** : Implémenter des rôles et des autorisations spécifiques (employé, manager, RH).
- **Soumettre une demande de congé** : Créer un formulaire de demande avec la possibilité de sélectionner le type et les dates.
- **Validation des demandes** : Permettre au manager de valider ou refuser une demande.

- **Consultation des congés** : La RH peut voir une liste des congés de tous les employés.

Employé - DemandeDeCongé :

Un employé peut soumettre plusieurs demandes de congé. (1:N) Chaque demande de congé appartient à un seul employé. (N:1) Manager -

Employé :

Un manager peut superviser plusieurs employés. (1:N)

Chaque employé a un seul manager. (N:1) Manager -

DemandeDeCongé :

Un manager peut valider plusieurs demandes de congé. (1:N) Chaque demande de congé est validée par un seul manager. (N:1) RH -

DemandeDeCongé :

Un responsable RH peut consulter plusieurs demandes de congé. (1:N)

Chaque demande de congé peut être consultée par un seul responsable RH. (N:1) Congé -

TypeDeCongé :

Un congé est de un type (par exemple, RTT, Vacances, etc.). (N:1)

Un type de congé peut être associé à plusieurs congés. (1:N) DemandeDeCongé -

Congé :

Une demande de congé peut être validée sous forme de congé. (1:1)

Chaque congé est le résultat d'une demande de congé validée. (1:1)

4.Code de base pour l'implémentation d'une demande de congé dans une interface :

CODE COMPLET INTERFACE voir fichier python annexe