



Test Technique pour le Poste de Tech Lead .NET/C#/Angular

Contexte

Vous devez implémenter un système de gestion des demandes de congé pour une entreprise. Vous travaillerez sur trois user stories décrites ci-dessous. Vous devrez démontrer votre capacité à concevoir une architecture robuste, développer les APIs, et écrire des tests unitaires.

User Stories

User Story 1 : Soumettre une demande de congé

Description :

En tant qu'employé, je veux soumettre une demande de congé via l'application RH afin de planifier mes vacances ou d'autres absences.

Critères d'acceptation :

1. Pour faire une demande de congé, l'employé doit renseigner les dates de début et de fin du congé, le type de congé (vacances, maladie, etc.), et un champ de commentaires optionnel.
2. Le formulaire doit valider que les dates de début et de fin sont correctes (par exemple, la date de fin ne peut pas être antérieure à la date de début).

User Story 2 : Approuver ou rejeter une demande de congé

Description :

En tant que gestionnaire RH, je veux approuver ou rejeter les demandes de congé afin de gérer efficacement les absences des employés.

Critères d'acceptation :

1. Le gestionnaire doit pouvoir approuver ou rejeter la demande avec la possibilité d'ajouter un commentaire explicatif.
2. Le statut de la demande doit être mis à jour en temps réel dans le système et visible pour l'employé.



Exigences Techniques

1. Architecture Hexagonale et Domain Driven Design (DDD)

- Concevez une architecture en utilisant les principes de l'architecture hexagonale et de DDD.
- Divisez le code en couches claires (Domain, Application, Infrastructure).
- Utilisez des agrégats, des entités, des valeurs-objets, et des services de domaine où cela est approprié.

2. APIs RESTful

- Implémentez des APIs RESTful pour chaque user story.
- Assurez-vous que les endpoints sont correctement versionnés et documentés (utilisez Swagger/OpenAPI si nécessaire).

3. Validation et Gestion des Erreurs

- Implémentez une validation appropriée des entrées utilisateur.
- Gérez les erreurs de manière cohérente et renvoyez des réponses d'erreur claires et informatives.

4. Tests Unitaires

- Écrivez des tests unitaires pour les composants critiques de votre application.
- Assurez-vous que vos tests sont bien organisés et couvrent les cas d'utilisation principaux.

Livrables

- **Code Source** : Un repository Git avec les instructions de construction et d'exécution.
- **Tests Unitaires** : Les tests unitaires avec une couverture suffisante des fonctionnalités principales.

Bonne chance 😊