

Phases du projet

1) Collecte des exigences :

L'objectif de ce projet est de créer un outil de gestion de projet (basé sur le web) pour les petites et moyennes équipes. Il y aura trois classes d'utilisateurs différentes :

- L'administrateur
- Le manager / chef de projet
- L'employé / développeur

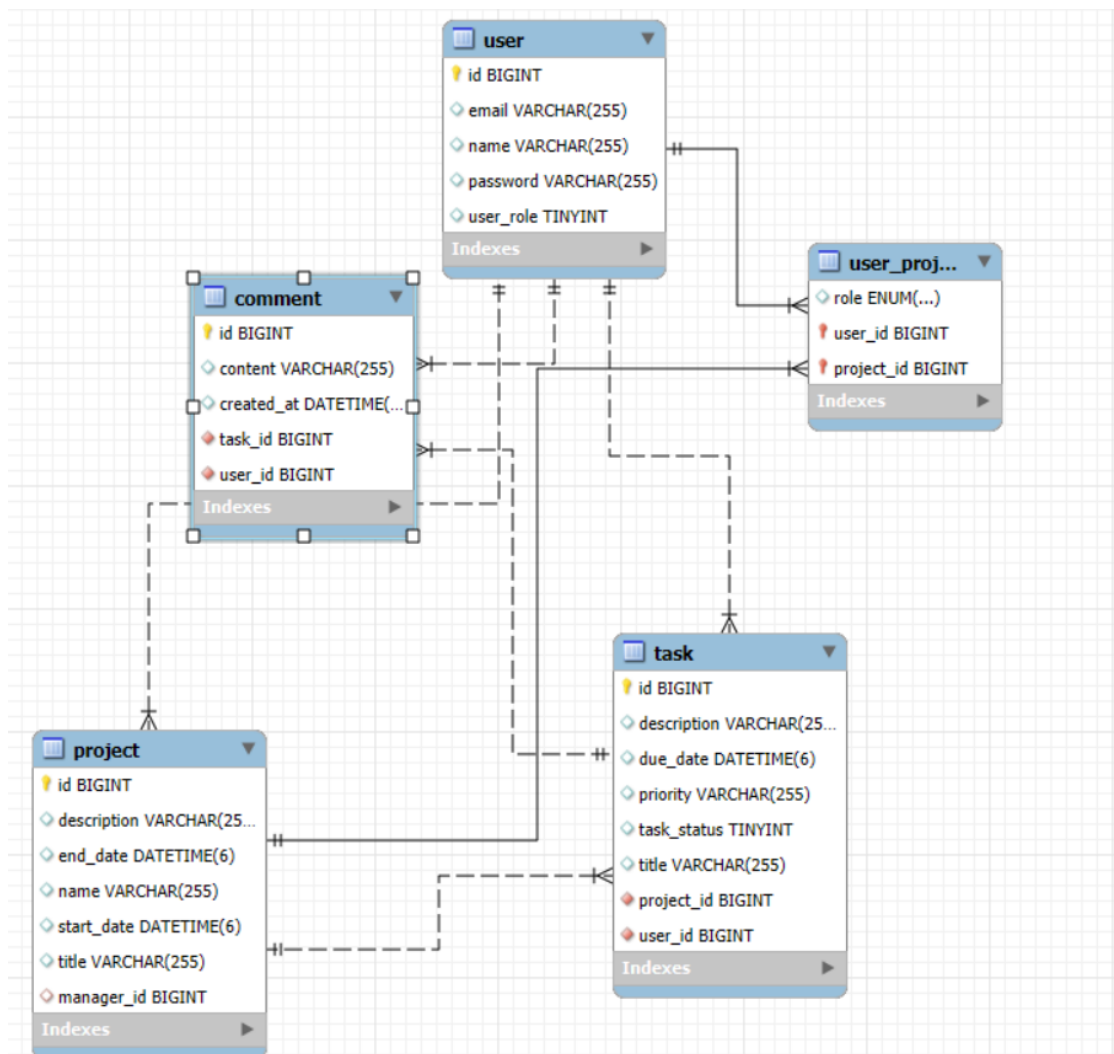
Ce projet devra répondre à des exigences tel que :

1. La connexion : les utilisateurs doivent pouvoir s'inscrire et se connecter ainsi que se déconnecter et gérer leur profil. Pour s'inscrire il faut renseigner son email, son nom et son mot de passe. Avec la création de compte vient un rôle (employé). L'administrateur sera celui qui peut changer le rôle d'un utilisateur d'employé à manager et inversement.
2. Un tableau de bord : le tableau de bord du manager fournira un aperçu des projets ainsi que des tâches. La progression des projets sera indiquée ainsi que le statut des tâches (en attente, en cours, complétée, etc..).
Le tableau de bord de l'employé fournira un aperçu des tâches auxquelles il est assigné.
3. Gestion de projet : Le manager est celui qui peut créer des projets et des tâches. Les projets ont un nom, une description, une date de début et de fin ainsi que la liste des membres. Lors de la création d'une tâche le manager indique l'échéance, la priorité, le nom, la description et à quel employé elle est assignée. Le manager pourra aussi modifier une tâche déjà créée (uniquement les tâches qu'il a lui-même créé).
Les employés pourront eux indiquer la complétion d'une tâche.
4. Collaboration d'équipe : possibilité d'ajouter des commentaires sur une tâche pour apporter une précision, une aide supplémentaire.
5. Notifications et alertes : les managers et les employés sont notifié à l'approche d'une échéance (date de fin de projet..).

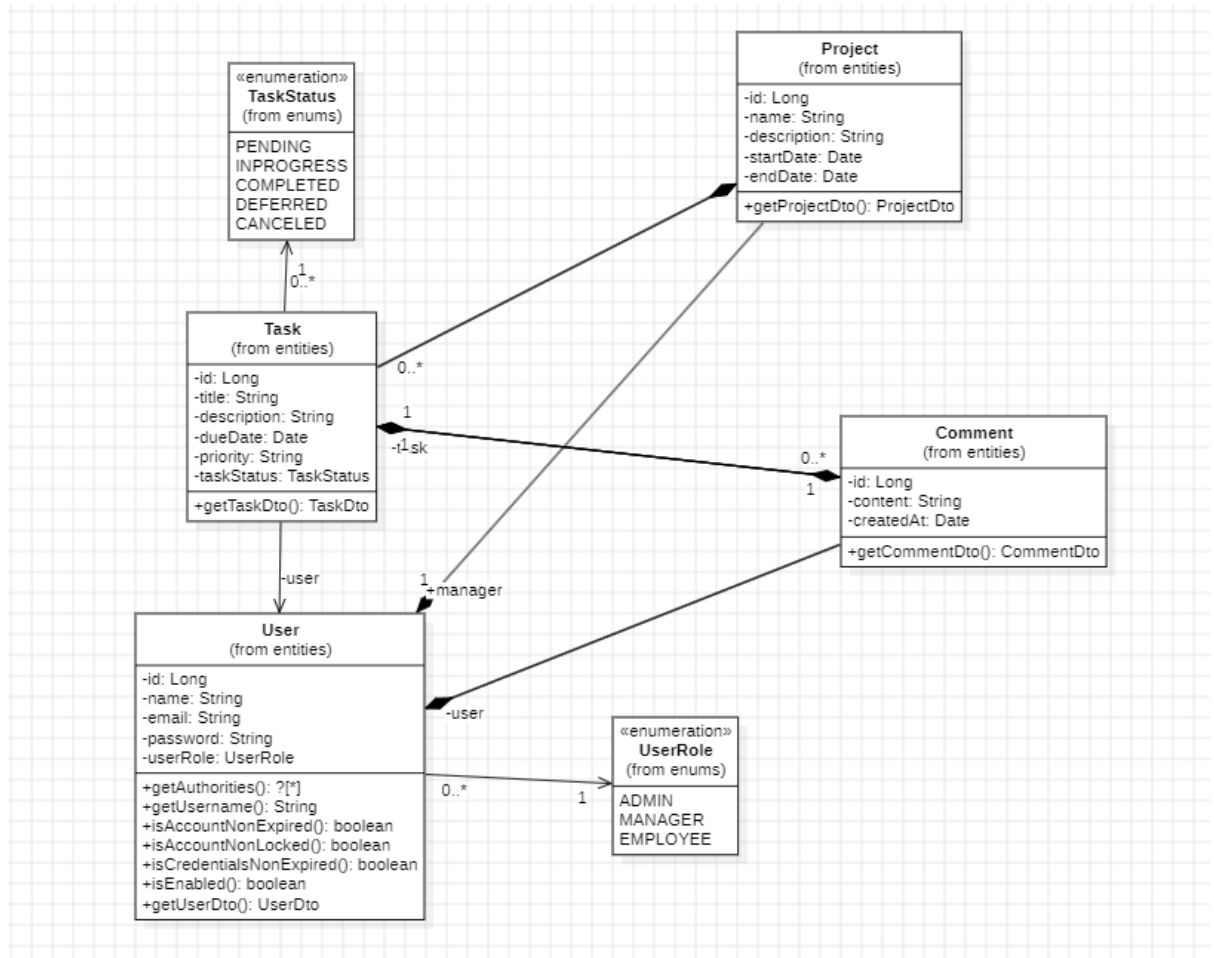
Il devra aussi répondre à des exigences non fonctionnelles. C'est à dire que le système doit être performant (fonctionne de la même façon même lorsqu'il y a beaucoup de monde connecté, etc..). Il doit être fiable, fonctionner de manière cohérente et avoir une tolérance aux erreurs. Il doit aussi être sécurisé (contrôle d'accès, sécurité des

L'implémentation se fera en Java pour le backend et on utilisera Angular pour le frontend.

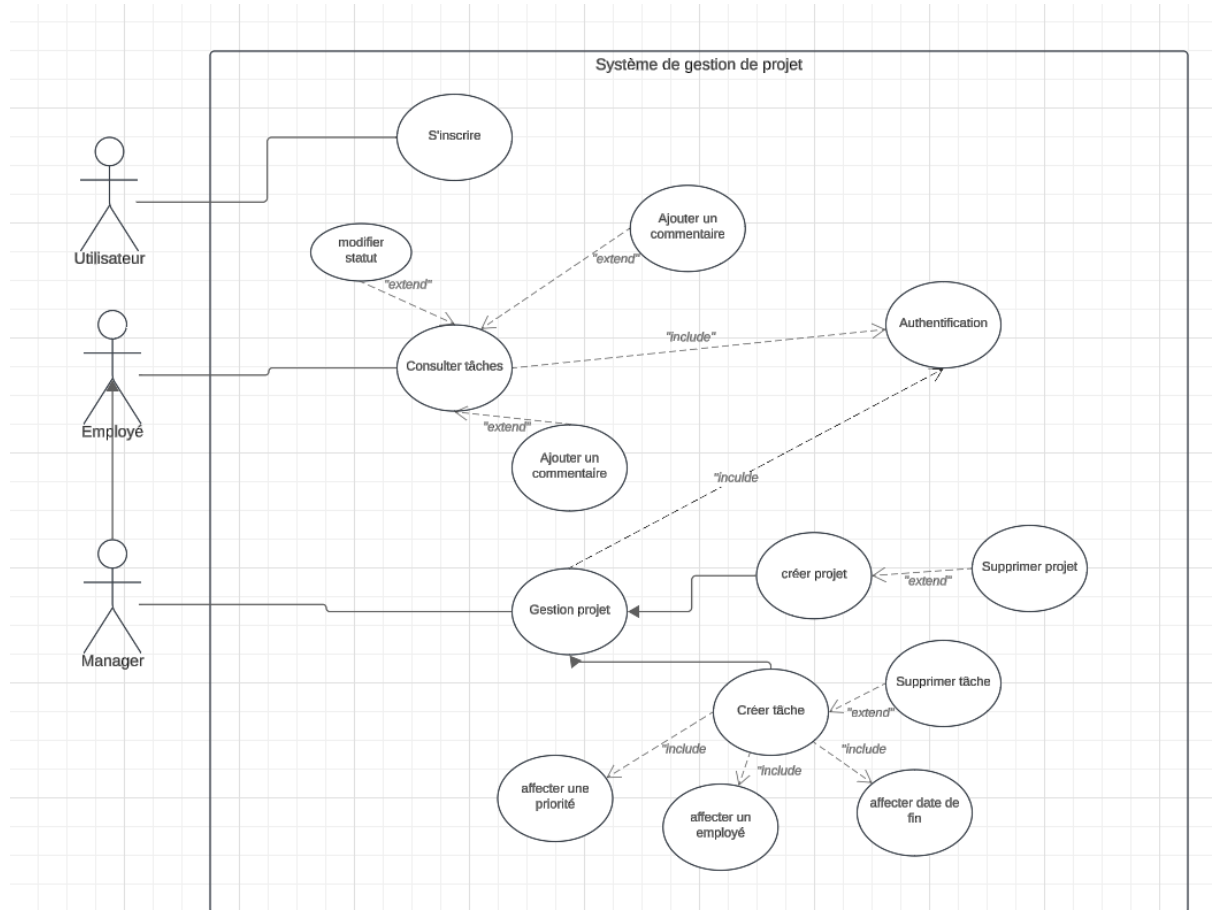
A. Schéma de la base de données :



B. Diagramme de classes :



C. Diagramme de cas d'utilisation :



L'architecture de notre système est plutôt orienté MVC (Modèle/Vue/Controller) puisqu'on a les controller qui font appel à leur classes service respectives (par exemple on a un managerController qui utilise ManagerService) comme ça les controller ont accès et modifient que les données qui leur sont liées. Donc les classes services remplissent le rôle de modèle. Pour la vue on a choisi d'utilisé angular.