

1. Présentation générale du projet

Contexte : Le projet concerne la gestion centralisée des activités de recherche et développement (R&D) pour TechInnova. L'objectif est de faciliter la gestion des projets, le suivi des idées innovantes, la collaboration, et la gestion des documents associés.

Technologies ciblées :

- Front-end : HTML5, CSS3, JavaScript (optionnel : React ou Vue.js).
 - Back-end : PHP avec MySQL ou PostgreSQL.
 - Sécurité : Authentification sécurisée et gestion des permissions.
-

2. Approche Merise

La méthode Merise permet de structurer et modéliser le système d'information en plusieurs étapes :

a) Niveau conceptuel : Modèle conceptuel des données (MCD)

Le **Modèle Conceptuel de Données (MCD)** représente graphiquement les entités et leurs relations. En voici une description simplifiée :

- **Projets** : Contient des informations sur les projets (titre, description, état...).
- **Idées** : Liées à des utilisateurs qui les soumettent et à des projets si acceptés.
- **Tâches** : Reliées à un projet avec un état d'avancement.
- **Documents** : Organisés par projet.
- **Utilisateurs** : Possèdent des rôles spécifiques (administrateur, collaborateur...).
- **Commentaires** : Reliés à leur auteur et appréciables par d'autres utilisateurs

Voici une description détaillée des relations dans notre MCD :

- ✓ **Projets et Tâches** :
 - Un projet peut avoir plusieurs tâches, mais chaque tâche appartient à un seul projet.
(1,n)
- ✓ **Projets et Documents** :
 - Un projet peut avoir plusieurs documents, et chaque document est associé à un seul projet.
(1,n)
- ✓ **Projets et Idées** :
 - Une idée peut être liée à zéro ou plusieurs projets.
(0,n)

✓ **Idées et Utilisateurs :**

- Chaque idée est soumise par un utilisateur, mais un utilisateur peut soumettre plusieurs idées.

(1,n)

✓ **Tâches et Projets, Documents et Projets, etc. :**

- Elles sont déjà établies, mais méritent un libellé clair.

✓ **Projets et Commentaires :**

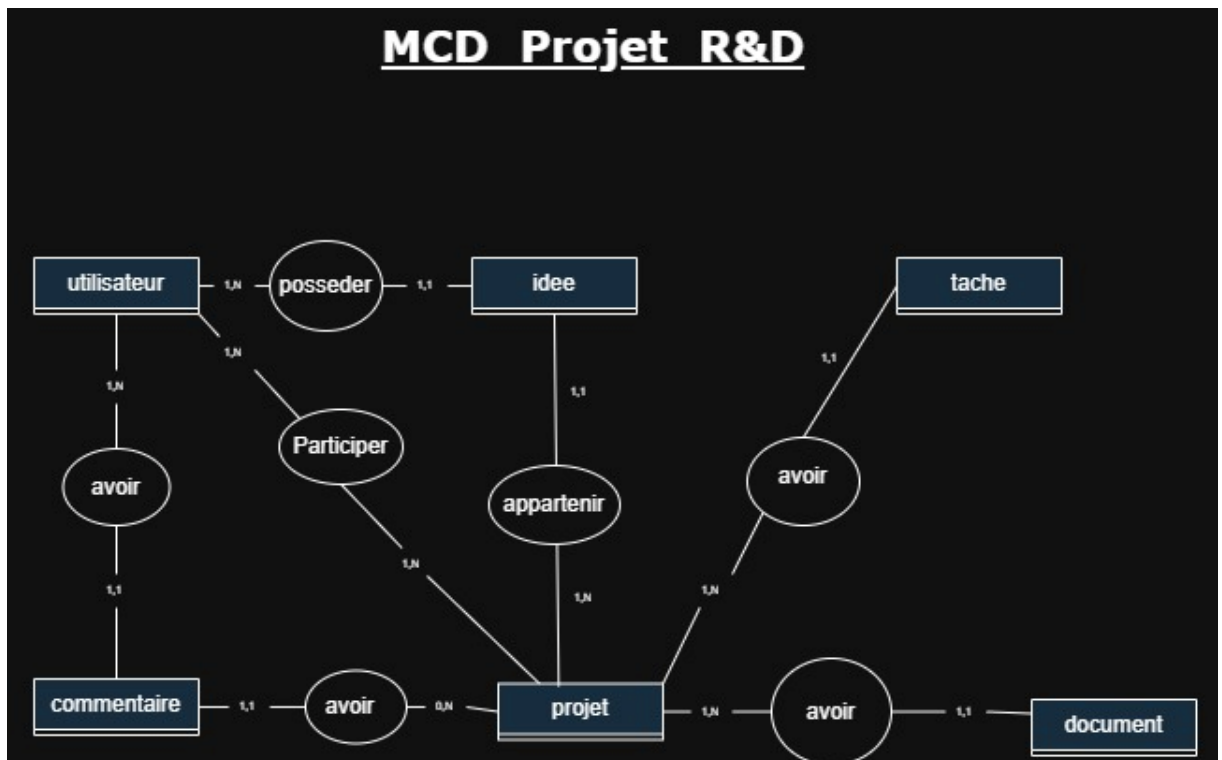
- Un projet peut avoir plusieurs commentaires, mais chaque commentaire appartient à un seul projet.

(1,n)

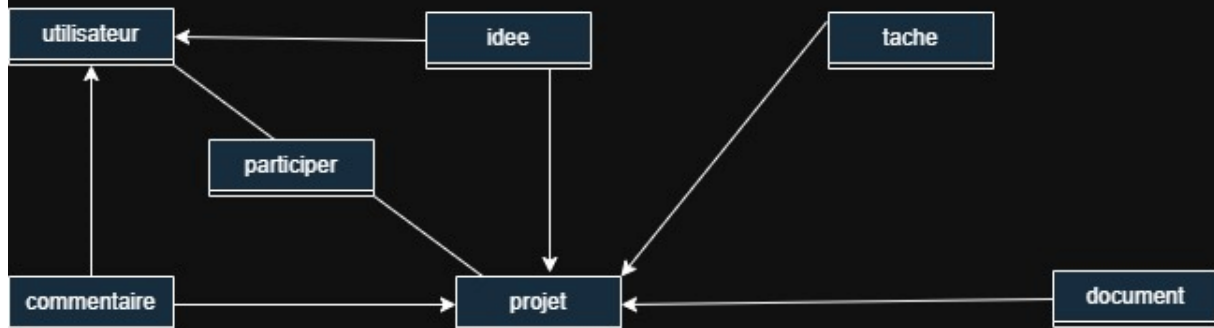
✓ **Utilisateurs et commentaires :**

- Un utilisateur peut avoir plusieurs commentaires, mais chaque commentaire appartient à un seul utilisateur.

(1,n)



MLD Projet R&D



<div> <div>utilisateur</div> <div> -id_utilisateur -nom -email -mot_de_passe -role -image_profile </div> </div>	<div> <div>commentaire</div> <div> -id_commentaire -auteur(fk_id_utilisateur) -proj_assoc(fk_id_projet) -contenu -date_emission </div> </div>	<div> <div>projet</div> <div> -id_projet -libelle -description -objectis -date_debut -date_fin -image_projet -etat </div> </div>	<div> <div>tache</div> <div> -id_tache -libelle -description -etat -date_emission </div> </div>
<div> <div>participer</div> <div> -id_participer -utilisateur_assoc(fk_id_utilisateur) -proj_assoc(fk_id_projet) </div> </div>	<div> <div>idee</div> <div> -id_idee -libelle -description -objectis -etat -date_emission -proj_assoc(fk_id_projet) </div> </div>		<div> <div>document</div> <div> -id_document -titre -description -fichier -proj_assoc(fk_id_projet) -date_emission -autorisation </div> </div>

Relations principales :

- Un projet peut contenir plusieurs tâches et documents.
- Une idée peut être soumise par un ou plusieurs utilisateurs.
- Un utilisateur peut être associé à plusieurs projets.

b) Niveau logique : Modèle logique des données (MLD)

Voici un extrait des tables principales en base de données :

- i. **Table projets :**
 - id (PK), titre, description, progression, etat, echeance.
 - ii. **Table idees :**
 - id (PK), titre, description, etat, utilisateur_id (FK vers utilisateurs).
 - iii. **Table utilisateurs :**
 - id (PK), nom, email, role.
 - iv. **Table documents :**
 - id (PK), nom, type, projet_id (FK vers projets).
 - v. **Table taches :**
 - id (PK), intitule, etat, projet_id (FK vers projets).
-

3. Plan d'architecture fonctionnelle

a) Front-end :

- Pages principales :
 - i. **Dashboard** : Résumé des projets, alertes, boutons pour ajouter des projets ou des idées.
 - ii. **Gestion des projets** : Liste des projets, filtres, et détails.
 - iii. **Soumission d'idées** : Formulaire pour ajouter une idée avec suivi de son état.
 - iv. **Gestion des utilisateurs** : Gestion des rôles et permissions pour l'admin.
 - v. **Documents** : Téléchargement et organisation selon les permissions.

b) Back-end :

- Gestion des rôles et permissions avec sessions sécurisées.
 - API pour gérer les données entre le front-end et la base de données.
-

4. Analyse des contraintes

- **Sécurité :**
 - Chiffrement des mots de passe.
 - Protection des accès par rôle (lecture, modification).
 - **Performance :**
 - Indexation des bases pour gérer les recherches par projet ou utilisateur.
 - Cache pour des opérations récurrentes (ex. tableau de bord).
-

5. Indicateurs de réussite

- Centralisation complète des données R&D.
 - Simplification du suivi des projets.
 - Collaboration fluide via des fonctionnalités de commentaires et votes.
-

6. Livrables

- Rapport technique détaillant :
 - Analyse Merise (MCD, MLD).
- Application complète en :
 - HTML, CSS, JavaScript (optionnel : intégration React/Vue.js).
 - PHP avec MySQL

7. Structure des fichiers pour le développement

L'organisation recommandée pour les fichiers de l'application est la suivante :

```
ProjetR&D/
├── assets/                # Ressources front-end
│   ├── css/              # Feuilles de styles CSS
│   ├── js/               # Scripts JavaScript
│   └── images/           # Images et icônes
├── includes/             # Fichiers PHP communs (header, footer, fonctions)
├── templates/            # Pages réutilisables (HTML/PHP)
├── uploads/              # Répertoire pour les documents uploadés
├── index.php             # Page d'accueil ou tableau de bord
├── projets.php           # Page gestion des projets
├── idees.php             # Page gestion des idées
├── utilisateurs.php      # Page gestion des utilisateurs
├── config.php            # Configuration de la base de données
└── database.sql          # Script SQL pour créer la base
```