

TD-1 Clarté et Structuration pour le Cahier des Charges de l'Application de Gestion de Tâches

1. Clarté de la Rédaction et de la Présentation

La **clarté de la rédaction** est cruciale pour s'assurer que tous les membres de l'équipe de développement, ainsi que les parties prenantes, comprennent bien les objectifs, les fonctionnalités, et les contraintes du projet. Voici comment garantir une présentation claire pour votre projet :

Rédaction Claire et Accessible

- **Vocabulaire Simple et Précis** : Utilisez un langage simple, sans jargon technique inutile, afin que toutes les parties prenantes (y compris les non développeurs) puissent comprendre les exigences du projet. Par exemple, au lieu de "gestion des autorisations avec hiérarchisation des rôles", utilisez une formulation plus simple comme "les utilisateurs peuvent avoir des permissions de lecture ou d'édition sur un projet".
- **Exemples Illustratifs** : Lorsque des fonctionnalités complexes sont décrites, il est souvent utile de fournir des exemples concrets ou des cas d'utilisation qui illustrent comment ces fonctionnalités fonctionneront. Cela aide à clarifier les attentes. Par exemple, pour la gestion des tâches, vous pouvez expliquer avec un exemple comment un utilisateur crée une tâche, la modifie, puis la marque comme terminée.

Cohérence dans les Termes Utilisés

- **Utilisation Uniforme des Terminologies** : Lorsque vous introduisez un terme spécifique (comme "projet", "groupe de tâches", "état"), assurez vous d'utiliser ces termes de manière cohérente tout au long du document pour éviter toute confusion.

Organisation Visuelle du Document

- **Titres et Sous-Titres** : Utilisez des titres et sous-titres pour structurer chaque section du cahier des charges de manière claire et identifiable. Chaque partie du document doit être bien délimitée :
 - **Introduction**
 - **Objectifs**
 - **Spécifications fonctionnelles**
 - **Spécifications techniques**

- **Sécurité et authentification**
 - **Synchronisation des données**
 - **Conclusion**
 - **Table des Matières** : Inclure une table des matières facilite la navigation dans le document. Chaque section doit être numérotée et facilement repérable.
 - **Formats de Mise en Page** : Utilisez du **gras** et de l'**italique** pour souligner les points importants. Les listes à puces (•) et les numérotations (**1. 2. 3.**) permettent de clarifier les étapes ou les exigences dans les fonctionnalités.
-

2. Structuration Logique des Spécifications

La **structuration logique des spécifications** permet de s'assurer que le cahier des charges suit un flux naturel, rendant chaque aspect du projet facile à comprendre. Voici les bonnes pratiques pour structurer le cahier des charges de ce projet de gestion de tâches :

Séparation des Spécifications Fonctionnelles et Techniques

- **Spécifications Fonctionnelles** : Elles doivent être clairement séparées des spécifications techniques. Les spécifications fonctionnelles concernent ce que l'application doit faire (par exemple, la création de projets, l'ajout de tâches, l'invitation de membres), tandis que les spécifications techniques définissent comment ces fonctionnalités seront mises en œuvre (technologies utilisées, architecture, etc.).

Organisation des Fonctionnalités en Catégories

- **Gestion des Projets** : Décrivez toutes les fonctionnalités relatives aux projets dans une section distincte. Cela inclut la création, modification, suppression, duplication de projets, ainsi que la gestion des permissions d'accès pour les membres.
- **Gestion des Groupes de Tâches et Tâches** : Placez toutes les fonctionnalités liées à la gestion des tâches et des groupes de tâches (création, modification, suppression, état des tâches) dans une section dédiée.
- **Sécurité et Authentification** : Toutes les fonctionnalités et mesures de sécurité, telles que la gestion des accès et l'authentification des utilisateurs, doivent être regroupées dans une section distincte.
- **Synchronisation des Données** : La manière dont les données sont synchronisées entre utilisateurs (collaboration en temps réel, mise à jour automatique des listes) doit être organisée dans une section spécifique.

Identification des Cas d'Utilisation

Organiser les spécifications en fonction des **cas d'utilisation** (user stories) permet de structurer les fonctionnalités autour des actions des utilisateurs. Par exemple :

- **Création d'un nouveau projet** : L'utilisateur entre un nom, une description, et choisit l'état du projet.
- **Ajout d'une tâche à un groupe de tâches** : L'utilisateur sélectionne le groupe de tâches, crée une tâche, la décrit, et attribue un état (à faire, en cours, terminé).

Chaque cas d'utilisation doit être clairement décrit et suivi des fonctionnalités associées.

Progression Logique

- **Introduction Générale du Projet** : Commencez par introduire l'objectif de l'application, les parties prenantes, et un aperçu général des fonctionnalités. Cela permet de poser le contexte avant d'entrer dans les détails techniques.
- **Description des Fonctionnalités de Base** : Développez les fonctionnalités essentielles comme la gestion des projets et des tâches, avant d'aborder les fonctionnalités avancées (duplication de projets, gestion des permissions).
- **Aspects Techniques et Sécurité** : Une fois les fonctionnalités décrites, passez à l'aspect technique : langages de programmation, base de données, sécurité. Cela montre comment les fonctionnalités seront concrétisées.

Tableaux, Schémas et Diagrammes

- **Diagrammes de Modèles de Données** : Un diagramme illustrant les relations entre les entités de la base de données (Users, Projects, Task Groups, Tasks) permet de clarifier la structure technique de l'application.
- **Tableaux Fonctionnalités/Permissions** : Un tableau décrivant les permissions (lecteur/éditeur) pour chaque utilisateur dans les projets permet d'organiser et de visualiser ces autorisations plus clairement.

Exemple de Structuration :

1. Introduction

- Objectif du Projet
- Parties Prenantes
- Périmètre Fonctionnel

2. Spécifications Fonctionnelles

- Gestion des Projets
- Gestion des Groupes de Tâches et Tâches

- Gestion des États
- Synchronisation des Données
- Sécurité et Permissions

3. Spécifications Techniques

- Choix du Langage de Programmation
- Architecture Backend et Frontend
- Modèle de Base de Données

4. Cas d'Utilisation

- Création d'un Projet
- Ajout de Membres à un Projet
- Création d'une Tâche

5. Mesures de Sécurité

- Gestion des Données Utilisateur
- Chiffrement des Mots de Passe
- Authentification des Utilisateurs

Cette structuration garantit une progression logique, une présentation claire, et une séparation des différents aspects du projet.