

TD-1 Gestion du Temps, Comment les 2 Semaines Sont Gérées et Utilisées

Le développement de l'application doit être réalisé dans un délai de **2 semaines**, un laps de temps limité qui nécessite une gestion optimale. Pour s'assurer que chaque aspect du projet soit bien traité dans ce délai, voici une répartition et une gestion stratégique du temps :

1. Planification Globale sur 2 Semaines

Pour garantir l'efficacité, le projet sera divisé en plusieurs phases avec des objectifs clairs pour chaque étape. Voici une proposition de répartition du temps :

- **Semaine 1** : Phase de conception et début du développement
 - **Jour 1-2** : Finalisation des spécifications fonctionnelles et techniques
 - **Jour 3-5** : Développement des fonctionnalités principales (gestion des projets et des tâches)
- **Semaine 2** : Finalisation du développement, tests et ajustements
 - **Jour 6-9** : Implémentation des fonctionnalités secondaires (sécurité, synchronisation)
 - **Jour 10-12** : Tests fonctionnels, ajustements, et correction de bugs
 - **Jour 13-14** : Révisions finales, validation et livraison du projet

2. Détail des Phases du Projet

Phase 1 : Conception et Finalisation des Spécifications (Jour 1-2)

- **Objectif** : Finaliser et valider les spécifications fonctionnelles et techniques du projet.
- **Actions** :
 - Rédaction et validation du cahier des charges avec l'équipe.
 - Révision des fonctionnalités de gestion des projets, des tâches, et des groupes de tâches.
 - Préciser la base de données, les technologies utilisées, et les méthodes d'authentification.
- **Temps estimé** : 2 jours pour une description complète et un alignement entre les membres de l'équipe de développement et les parties prenantes.

Phase 2 : Développement des Fonctionnalités Principales (Jour 3-5)

- **Objectif** : Commencer le développement des fonctionnalités essentielles de l'application.
- **Actions** :
 - Création de l'interface utilisateur pour la gestion des projets et des listes de tâches.
 - Développement des fonctionnalités permettant la création, modification et suppression des projets.
 - Développement de la structure backend pour gérer les groupes de tâches et les états des tâches.
- **Temps estimé** : 3 jours pour le développement des fonctionnalités principales.

Phase 3 : Implémentation des Fonctionnalités Secondaires (Jour 6-9)

- **Objectif** : Ajouter les fonctionnalités avancées telles que la sécurité, la synchronisation, et la gestion des permissions.
- **Actions** :
 - Mise en place du système de sécurité et d'authentification des utilisateurs (chiffrement des mots de passe, gestion des accès).
 - Gestion des permissions (lecteur/éditeur) et implémentation des rôles d'utilisateur.
 - Implémentation de la synchronisation des données entre utilisateurs en temps réel (collaboration sur les listes de tâches).
- **Temps estimé** : 4 jours pour développer et tester ces fonctionnalités complexes.

Phase 4 : Tests, Corrections de Bugs et Finalisation (Jour 10-12)

- **Objectif** : Effectuer des tests rigoureux sur les fonctionnalités développées et corriger les bugs.
- **Actions** :
 - Tests unitaires sur chaque fonctionnalité pour s'assurer qu'elles fonctionnent correctement.
 - Tests d'intégration pour vérifier la bonne interaction entre les différentes parties de l'application (projets, tâches, utilisateurs).
 - Révision des problèmes rencontrés lors des tests et correction des bugs critiques.
- **Temps estimé** : 3 jours pour une phase de tests approfondie et la correction de bugs.

Phase 5 : Validation Finale et Livraison (Jour 13-14)

- **Objectif** : Finaliser le projet, effectuer une dernière révision et le livrer.
 - **Actions** :
 - Révision finale du projet avec l'équipe, en s'assurant que toutes les fonctionnalités sont présentes et fonctionnent comme prévu.
 - Présentation du produit fini aux parties prenantes et prise en compte des derniers retours.
 - Livraison de l'application dans les délais.
 - **Temps estimé** : 2 jours pour effectuer une dernière vérification et finaliser la livraison.
-

3. Suivi et Ajustement du Calendrier

- **Réunions de Suivi** : Des réunions quotidiennes courtes (**stand-ups**) doivent être organisées pour suivre l'avancement du projet, identifier les obstacles, et ajuster le planning si nécessaire. Cela permet d'avoir une vue claire de l'avancement à chaque étape.
- **Méthode Agile** : Utiliser une méthodologie de développement agile en travaillant par sprints courts (chaque jour ou sur 2-3 jours) pour permettre des ajustements rapides en fonction des résultats de chaque phase.

4. Gestion des Imprévus

Il est important de prévoir une marge de manœuvre en cas d'imprévus, comme des difficultés techniques ou des retours inattendus des parties prenantes. Dans ce cas, des priorités devront être définies afin de livrer les fonctionnalités les plus critiques en premier, en laissant certaines fonctionnalités secondaires pour un développement postérieur si le temps manque.

Conclusion

Cette répartition du temps sur les deux semaines permet de s'assurer que chaque fonctionnalité et chaque aspect technique soit correctement implémenté, testé et finalisé dans les délais impartis. La rigueur dans le suivi du planning est la clé pour éviter tout retard et livrer une application fonctionnelle dans le respect des attentes du projet.