

Énoncé de Travail : Rédaction du Cahier des Charges - Développement d'une application web de recommandation des espaces de travail

Objectif :

Rédiger un cahier des charges technique et fonctionnel pour la conception d'une plateforme intelligente recensant et recommandant des lieux (cafés, bibliothèques, coworking) adaptés au télétravail ou aux révisions. Ce travail en binôme exige une maîtrise des requêtes bases de données dynamiques (moteur de recherche par tags), tout en vous laissant la liberté d'enrichir l'algorithme de classement et l'interface mobile.

Veuillez structurer votre document final en vous appuyant sur les axes suivants. Sentez-vous libres d'enrichir chaque section avec vos propres idées d'ingénierie :

1. Contexte, Paramètres et Objectifs du Projet

- **Tâche :** Rédigez une introduction expliquant la problématique actuelle (la difficulté de trouver le bon endroit pour travailler selon des besoins très précis) et comment votre plateforme va la résoudre.
- **Questions et Pistes de réflexion :**
 - En quoi cette application diffère-t-elle de Google Maps ou TripAdvisor dans son approche métier ?
 - Quelles sont les technologies fondamentales exigées (Angular, Spring Boot avec JPA Criteria API, MySQL, API LLM) ? Vous pouvez en proposer d'autres en complément (ex: intégration de l'API Google Maps pour la cartographie, Redis pour le cache des recherches fréquentes).
 - Quelle est la durée allouée (3 mois) et le volume horaire exigé ?

2. Définition des Acteurs du Système

- **Tâche :** Identifiez et définissez les rôles et permissions de l'Utilisateur (Travailleur/Étudiant), du Propriétaire (Gérant du lieu) et de l'Administrateur.
- **Questions et Pistes de réflexion :**
 - Quelles sont les actions réservées au Propriétaire concernant la mise à jour des informations de son établissement (ajout de tags/critères) ?
 - Comment l'Administrateur gère-t-il la base globale des critères (ex: ajout d'un nouveau tag "Vegan-friendly" sans avoir à redéployer le code) ?

- Réflexion ouverte : Serait-il pertinent d'ajouter un rôle de "Modérateur Communautaire" (des utilisateurs très actifs qui valident les modifications suggérées par d'autres) ?

3. Spécifications Fonctionnelles du Socle Core

- **Tâche :** Détaillez les processus liés à la gestion des lieux, au système d'avis et au moteur de recherche par attributs (Tags).
- **Questions et Pistes de réflexion :**
 - **Le Moteur de Recherche :** C'est le cœur du projet. Comment concevoir l'interface Angular pour que l'utilisateur puisse cocher/décocher dynamiquement des dizaines de critères (Wifi, Prises, Calme, Ouvert le dimanche) ?
 - **Système d'Avis et Ranking :** Décrivez comment l'application calcule la moyenne globale d'un lieu. Comment l'algorithme de classement trie-t-il les résultats pertinents en cas d'égalité sur les tags ?

4. Module Innovant : Fonctionnalité Avancée (Extension IA)

- **Tâche :** Pour démarquer votre projet, décrivez l'intégration d'un "Concierge Virtuel" (Smart Finder) utilisant l'IA pour la recherche en langage naturel.
- **Questions et Pistes de réflexion :**
 - Au lieu de cocher des cases, l'utilisateur tape : "Je cherche un café calme ouvert le dimanche avec des prises pour bosser". Comment le backend va-t-il utiliser l'API LLM pour extraire ces intentions et les transformer en une liste de Tags (IDs) compréhensibles par la base de données ?
 - Quel format de réponse JSON attendez-vous de l'IA pour faciliter cette conversion ?
 - Réflexion technique : Que doit répondre le système si l'IA extrait un critère qui n'existe pas encore dans votre base de données ?

5. Architecture, Modélisation et Requêtes Dynamiques

- **Tâche :** Proposer la structure de la base de données (MySQL) et expliquer le fonctionnement des requêtes de recherche dynamiques.
- **Questions et Pistes de réflexion :**

- **Modélisation (Many-to-Many)** : Listez les entités principales (Utilisateur, Lieu, Critere, Avis). Expliquez l'importance de la table de jointure entre Lieu et Critere.
- **JPA Specifications (Criteria API)** : Pourquoi est-il impossible d'utiliser des requêtes SQL statiques (ex: @Query) pour un moteur de recherche où l'utilisateur peut combiner aléatoirement 2, 5 ou 10 filtres différents ? Expliquez le concept de construction de requêtes dynamiques côté Spring Boot.

6. Exigences Non-Fonctionnelles et Techniques

- **Tâche** : Spécifier les contraintes techniques liées à l'interface mobile, aux performances et à la sécurité.
- **Questions et Pistes de réflexion** :
 - **Mobile-First (UI/UX)** : Ce type d'application est utilisé dans la rue sur smartphone. Pourquoi l'utilisation de frameworks comme Tailwind CSS ou Bootstrap est-elle critique ici ?
 - **Performances SQL** : La recherche avec plusieurs jointures (JOIN) peut être très lente. Pourquoi est-il crucial de créer des "Index" sur les tables MySQL ?
 - **Sécurité (JWT)** : Comment protéger strictement les routes de création de nouveaux critères (POST /api/criteres) pour qu'elles ne soient accessibles qu'à l'Admin ?

7. Planning Prévisionnel de Réalisation Extensible

- **Tâche** : Construire un diagramme ou un tableau de planification sur 12 semaines.
- **Questions et Pistes de réflexion** :
 - Intégrez impérativement les 5 phases clés (Modélisation & Socle, Le Moteur de Recherche, Ranking & Avis, IA Smart Finder, Optimisation & Recettage).
 - Travail demandé : Détaillez et enrichissez ce planning. Intégrez des étapes cruciales comme le maquettage UI Mobile-First, la réalisation de tests de charge sur la base de données (pour simuler des recherches complexes simultanées), et le déploiement.