

Consigne de projet – Sprint Agile pour C’Coworker

Durée du Sprint : 1 à 1,5 jour

Date de début : mardi

Présentation produit : lundi 2 février fin d’après-midi

L’objectif : développer les fonctionnalités principales de l’application **C’Coworker**, en appliquant toutes les bonnes pratiques de développement Agile.

1. Contexte et objectif du projet

C’Coworker est une entreprise proposant des **espaces de coworking et des salles de réunion à louer**.

Le projet consiste à créer un **site web pour le personnel administratif**, permettant de **gérer facilement les espaces et les réservations**.

Le site doit être **simple, clair et interactif**, avec des formulaires et des règles logiques pour éviter les erreurs.

Extension optionnelle : Le personnel administratif pourra se connecter via un compte créé et avoir accès à toutes les fonctionnalités. Les utilisateurs externes pourront réserver eux-mêmes un espace une fois la partie administrative fonctionnelle.

2. Public cible

Phase principale – Personnel administratif :

- Créer, modifier et supprimer des espaces et des réservations
- Consulter le planning et vérifier l’absence de doublons

Phase optionnelle – Utilisateurs externes :

- Réserver un espace ou une salle directement
-

3. Pages et fonctionnalités simplifiées

3.1 Gestion des espaces

- Liste des espaces : tous les bureaux et salles avec leur capacité et type (open-space/bureau/salle de réunion)
- Ajouter un espace
- Modifier ou supprimer un espace

3.2 Gestion des réservations

- Créer une réservation : choix de l’espace, date, heure de début et de fin, nom de l’utilisateur
- Modifier ou supprimer une réservation
- Vérification automatique des conflits

3.3 Planning et tableau de bord

- Planning hebdomadaire : voir les espaces et réservations par jour et heure

- Tableau de bord : aperçu des espaces libres/occupés et des prochaines réservations

3.4 Gestion des utilisateurs

- Liste des utilisateurs
 - Ajouter un utilisateur
 - Modifier ou supprimer un utilisateur
-

4. Logique du site

- Un espace ne peut pas être réservé deux fois au même moment
-

5. Interface et ergonomie

- Menus simples pour naviguer entre les pages
 - Codes couleur pour l'état des espaces (libre/occupé)
 - Tableaux et formulaires faciles à lire et à remplir
 - Messages clairs pour confirmer ou signaler une erreur
 - Fonctionne sur ordinateur et tablette (si le temps le permet)
-

6. Technique

- **Front-end** : HTML, CSS, JavaScript
 - **Back-end** : PHP
 - **Base de données** : MySQL
 - Vérification des informations côté client et côté serveur
-

7. Livrables attendus

- Code source complet et fonctionnel
 - Base de données prête à l'emploi
 - Documentation simple pour installation et usage (rapide au vu du temps réduit)
 - Site fonctionnel avec toutes les fonctionnalités principales
-

8. Critères de réussite

- Fonctionnalités administratives entièrement opérationnelles
- Pas de conflits dans les réservations
- Interface claire et simple à utiliser

- Respect du cahier des charges
-

9. Organisation Agile du Sprint

Rôles dans l'équipe :

| Rôle | Responsabilités |
|--------------------|---|
| Développeurs | Développement des fonctionnalités, tests, refactoring, Pull Requests |
| Lead Developer | Organisation des Code Reviews, validation technique des PR, support technique |
| Product Owner (PO) | Validation fonctionnelle, priorisation des tickets, vérification du respect des besoins métiers |
| Scrum Master | Facilitation des Dailies, suivi du Kanban, suivi du Burndown Chart, levée des blocages |

Notions Agile à appliquer dans le projet :

1. **Peer Programming** : développement en binôme pour partager les connaissances et améliorer la qualité du code
2. **TDD (Test Driven Development)** : écrire les tests avant le code pour chaque fonctionnalité
3. **KISS (Keep It Simple and Stupid)** : privilégier des solutions simples et lisibles
4. **Refactoring** : à la fin de chaque demi-journée, améliorer le code, renommer variables/fonctions, clarifier la logique
5. **Daily Scrum** : réunion quotidienne organisée par le Scrum Master
6. **Pull Request et Code Review** : validation technique par le Lead Developer et l'équipe
7. **Validation fonctionnelle** : chaque ticket validé par le Product Owner
8. **Kanban et Burndown Chart** : suivi continu de l'avancement et mise à jour après validation des tickets. (Scrum Master)

Fin du Sprint :

- **Sprint Review** : présentation du produit terminé au Product Owner et au client
- **Sprint Rétrospective** : discussion sur ce qui a bien fonctionné, difficultés rencontrées, pistes d'amélioration pour le prochain Sprint