

Progetto: Piattaforma “CoWorkSpace” per la gestione di spazi condivisi

1. Descrizione del Caso d'Uso

Scenario:

Un'azienda vuole creare una piattaforma digitale che consenta a professionisti e aziende di prenotare spazi di coworking in diverse città. Ogni sede offre stanze private, postazioni flessibili, sale riunioni e servizi accessori. Gli utenti devono poter esplorare le sedi, verificare disponibilità in tempo reale e prenotare in autonomia.

Obiettivi principali:

- **Catalogo sedi:** Visualizzazione delle sedi con filtri per città, tipologia di spazio, servizi e disponibilità.
- **Gestione account:** Registrazione, autenticazione e gestione del profilo utente.
- **Prenotazione e pagamento:** Prenotazione degli spazi, gestione del calendario e pagamento online.
- **Dashboard gestori:** Dashboard per i responsabili delle sedi per gestire disponibilità, prenotazioni e reportistica.
- **Deployment Cloud:** Deploy su piattaforma cloud (AWS, GCP, ecc.) per garantire scalabilità e continuità di servizio.

2. Intervista Simulata col Potenziale Cliente

Intervistatore:

"Ci racconti l'idea dietro questa piattaforma di coworking?"

Cliente (CoWorkSpace):

"Vogliamo rendere semplice la prenotazione di spazi professionali in tutta Italia. I clienti devono trovare rapidamente una sede, selezionare un tipo di spazio e prenotare con pochi click. Dall'altra parte, i gestori devono poter aggiornare disponibilità e ricevere notifiche in tempo reale."

Intervistatore:

"Chi saranno gli utenti principali della piattaforma?"

Cliente:

"Ci sono tre categorie di utenti:

1. **Clienti:** freelance, aziende, team in trasferta.
2. **Gestori:** responsabili delle sedi che aggiornano spazi e orari.
3. **Amministratori:** controllano l'infrastruttura, supportano utenti e gestori."

Intervistatore:

"Quali funzionalità ritenete indispensabili e quali sono le priorità tecniche?"

Cliente:

"La gestione dinamica del calendario, il sistema di pagamento sicuro, notifiche automatiche e un'interfaccia responsive. Il tutto su cloud per affrontare eventuali picchi di accesso."

Intervistatore:

"Avete preferenze sulla piattaforma di deployment?"

Cliente:

"Preferiamo una soluzione cloud scalabile, lasciando libertà tecnica al team: AWS, GCP o simili, purché ben documentata."

3. Divisione dei Ruoli e Tempistiche

Ruoli del Team

Frontend Developer(s)

- **Responsabilità:**
 - Progettazione dell'interfaccia utente per esplorazione sedi, prenotazioni e gestione profilo.
 - Comunicazione dinamica con backend (API) per visualizzare disponibilità e confermare prenotazioni.
- **Deliverables:**

- UI responsive HTML/CSS/JS e prototipi interattivi.
- Documentazione UI e test di compatibilità browser.

Backend Developer(s)

- **Responsabilità:**

- Creazione API RESTful (es. Node.js/Express) per gestione utenti, sedi, disponibilità e pagamenti.
- Autenticazione, sicurezza e notifiche automatiche (es. email o push).

- **Deliverables:**

- Codice backend e documentazione (OpenAPI/Swagger).
- Test automatici unitari e di integrazione.

Database Developer/Specialista

- **Responsabilità:**

- Modellazione di un database relazionale per utenti, sedi, disponibilità, prenotazioni e pagamenti.
- Script per setup iniziale, backup e gestione in cloud.

- **Deliverables:**

- Diagrammi ER, script SQL e guide CRUD.
- Documentazione gestione database in ambiente cloud.

DevOps/Infrastruttura

- **Responsabilità:**

- Deploy dell'applicazione in ambiente cloud.
- Configurazione pipeline CI/CD, monitoraggio e scaling automatico.

- **Deliverables:**

- Script per deploy (Dockerfile, workflow CI/CD).
 - Documentazione deploy e gestione ambienti.
-

4. Modalità di Consegna

- **Repository Git:**

- Branch separati per frontend, backend, database e setup DevOps.
- README completo con istruzioni per deploy e configurazione ambienti cloud.

- **Documentazione Tecnica:**

- Specifiche API, schema ER, istruzioni deploy, gestione errori e rollback.

- **Demo:**

- Video o sessione live dimostrativa della piattaforma funzionante in cloud.

- **Testing:**

- Integrazione di test automatici per validazione delle funzionalità e verifica di scalabilità.
-

5. Criteri di Valutazione

- **Funzionalità:**

- Gestione completa di prenotazioni, sedi, utenti e dashboard.

- **Qualità del Codice:**

- Standard di scrittura, modularità e uso corretto di pattern.

- **UI/UX:**

- Interfaccia responsive, intuitiva e accessibile.
- **Integrazione e Sicurezza:**
 - Coordinamento tra componenti e attenzione alla protezione dei dati e dei pagamenti.
- **Deploy e Scalabilità:**
 - Deploy in cloud riuscito, con capacità di scaling e monitoraggio.
- **Documentazione e Testing:**
 - Completezza di documentazione e qualità dei test automatici.