

1) Contexte du projet

développer une application console Java de gestion de réservations d'hôtels permettant la création et la gestion des hôtels, la réservation et l'annulation de chambres, ainsi que la consultation des disponibilités et de l'historique des réservations.

1.1 Cible : apprenants Java (POO), bonnes pratiques (encapsulation, collections, validation des données).

1.2 Vision : produire un socle propre, testable et facile à comprendre pour un système de réservation hôtelier.

2) Objectifs pédagogiques

- Mettre en œuvre la POO (entités Hotel, Reservation, Client, services, repositories, séparation des responsabilités).
- Manipuler des données numériques et textuelles avec validation (nombre de chambres, nuits réservées, email).
- Implémenter une authentification basique (email + mot de passe) pour sécuriser les réservations et actions sensibles.
- Appliquer des règles métier (disponibilité des chambres, annulation, réservation par client).
- Produire un code structuré, lisible et testable.

3) Périmètre fonctionnel

3.1 Authentification & Profil

- Inscription (Register) : nom, prénom, email (unique), mot de passe (min 6 caractères).
- Connexion (Login) : email + mot de passe.
- Session : l'utilisateur connecté peut accéder à ses réservations et hôtels ; déconnexion possible.
- Mise à jour profil : modification de l'email ou mot de passe (contrôle de forme).

3.2 Hôtels

- Création d'hôtel : identifiant unique, nom, adresse, nombre de chambres disponibles, note.
- Liste des hôtels : consultation globale ou filtrée (par disponibilité).
- Mise à jour : modification du nom, adresse, nombre de chambres.
- Suppression d'hôtel : autorisée si aucune réservation en cours.

3.3 Réservations

- Réserver une chambre : décrémentation automatique des chambres disponibles.

- Annuler une réservation : incrémentation du nombre de chambres disponibles.
- Historique des réservations : tri chronologique, affichage détaillé (date, client, nuits, hôtel).

3.4 Interface console

- Menus clairs, messages d'erreur explicites, saisie guidée.
- Validation des entrées : nombre de chambres > 0, email non vide, mot de passe ≥ 6 caractères.

4) Rôles & Acteurs

- Utilisateur authentifié : crée et annule des réservations, consulte son historique et modifie son profil.
- Administrateur (optionnel) : crée, modifie ou supprime des hôtels.

5) Règles métier (exigences)

- Disponibilité : réservation refusée si aucune chambre libre.
- Annulation : possible seulement si réservation existante.
- Propriété : un utilisateur ne peut annuler que ses propres réservations.
- Traçabilité : toute réservation ou annulation est enregistrée dans l'historique.
- Unicité email : l'inscription échoue si l'email existe déjà.

6) Modèle conceptuel (domain)

Client

- id: UUID, fullName: String, email: String, password: String

Hotel

- hotelId: String, name: String, address: String, availableRooms: int, rating: double

Reservation

- id: UUID, timestamp: Instant, hotelId: String, clientId: UUID, nights: int

Invariants

- availableRooms ≥ 0 pour autoriser la réservation.
- nights > 0 pour toute réservation.

7) Architecture logicielle

- Main (UI console) : menus, saisies, affichages, contrôle du flux.
- Services :

- o AuthService : inscription, connexion, profil, mot de passe.
- o HotelService : création, modification, suppression, liste, disponibilité.
- o ReservationService : réservation, annulation, consultation historique.
- Repositories :
- o Interfaces : ClientRepository, HotelRepository, ReservationRepository.
- o Implémentations en mémoire : InMemoryClientRepository, etc.
- Utils : saisie console, validation des entrées.

8) Parcours utilisateur (console)

Accueil (non connecté)

- Register
- Login
- Exit

Connecté (« Logged in as ... »)

- Create hotel (admin)
- List hotels
- Reserve room
- Cancel reservation
- Reservation history
- Update profile
- Change password
- Logout
- Exit

9) Résultats attendus

- Inscription/connexion fonctionnelles ; unicité de l'email contrôlée.
- Création/liste/modification/suppression d'hôtels conformes aux règles.
- Réservations/annulations : mise à jour du nombre de chambres + enregistrement dans l'historique.
- Historique trié par date avec détails complet.

- Messages d'erreur clairs (chambres indisponibles, réservation inexistante, non-propriétaire, etc.).