Teamwork 5 – Report

Team name: OpsForge

Team members: Mao Tamura (e2502119), Dominik Stiegler (e2502128), Dario Mathys(e2502127)

This class diagram models a hotel reservation system. The main entities are Hotel, Room, Reservation, Guest, and supporting services and repositories.

- **Hotel and Rooms**:
  A Hotel contains many Rooms (1…*). Each Room has attributes such as roomNumber, type, capacity, and status. Room types (SINGLE, DOUBLE, SUITE, FAMILY) and statuses (ACTIVE, MAINTENANCE) are defined by enums.
- **Reservations**:
  A Reservation links a Guest and a Room. It includes details such as checkIn, checkOut, and a status (PENDING, CONFIRMED, etc.). Each Reservation uses a **DateRange** object to represent its duration, with methods to check overlap or conflicts. A Guest may have multiple Reservations.
- **Services**:
  The **ReservationService** handles creating, canceling, updating reservations, and searching available rooms. It depends on both the **RoomRepository** and the **ReservationRepository**.
  The **AvailabilityService** checks room availability for given date ranges, using the ReservationRepository. ReservationService also uses AvailabilityService.
- **Repositories**:
  RoomRepository and ReservationRepository are interfaces for data persistence. They provide methods for retrieving, saving, and deleting Rooms or Reservations.
- **Interactions**:
  - The Hotel manages its Rooms (add, remove, get).
  - A Guest books a Room through ReservationService, which checks availability via AvailabilityService before saving to ReservationRepository.
  - DateRange ensures no overlapping reservations.
  - ReservationStatus tracks the lifecycle of a booking.

Summary:
The system separates core entities (Hotel, Room, Guest, Reservation) from service logic (ReservationService, AvailabilityService) and persistence (repositories). This design improves clarity, supports extensibility, and enforces clear responsibilities.