

Database Model Design

This document outlines the database models used in our FlyNext project. The design is based on the following models:

User: Represents both regular users and hotel owners. Contains personal details and authentication information. Each user can have multiple bookings and notifications.

Hotel: Represents a hotel listed on the platform. Each hotel can have multiple rooms and bookings.

Room: Represents a room type in a hotel. It includes details like amenities, price per night, and the number of available rooms. Each room belongs to a hotel and can be booked multiple times.

Booking: Represents a hotel booking which represents a hotel reservation. It links users with hotels and rooms.

FlightBooking: Represents flight reservations made through the AFS API. Stores minimal details and is maintained in a separate table to distinguish flight bookings from hotel bookings.

Notification: Represents messages sent to users regarding the status of their bookings or other important updates.

City and Airport: These models store location and airport data fetched from the AFS API.

The following diagram (see attached ER diagram on next page) illustrates the relationships between these models. Notice that:

- A **User** can create multiple **Bookings/FlightBookings** and receive multiple **Notifications**.
- A **Hotel** is owned by a single **User** and can offer multiple **Rooms** and receive multiple **Bookings**.
- A **Room** belongs to one **Hotel** and can be associated with multiple **Bookings**.

The ER diagram provides additional details on the exact columns, data types, and constraints for each table.

Note on Enum Fields:

- **UserRole:** Possible values are **USER** and **HOTEL_OWNER**.
- **BookingStatus:** Possible values are **CONFIRMED**, **CANCELED**, and **PENDING**.

