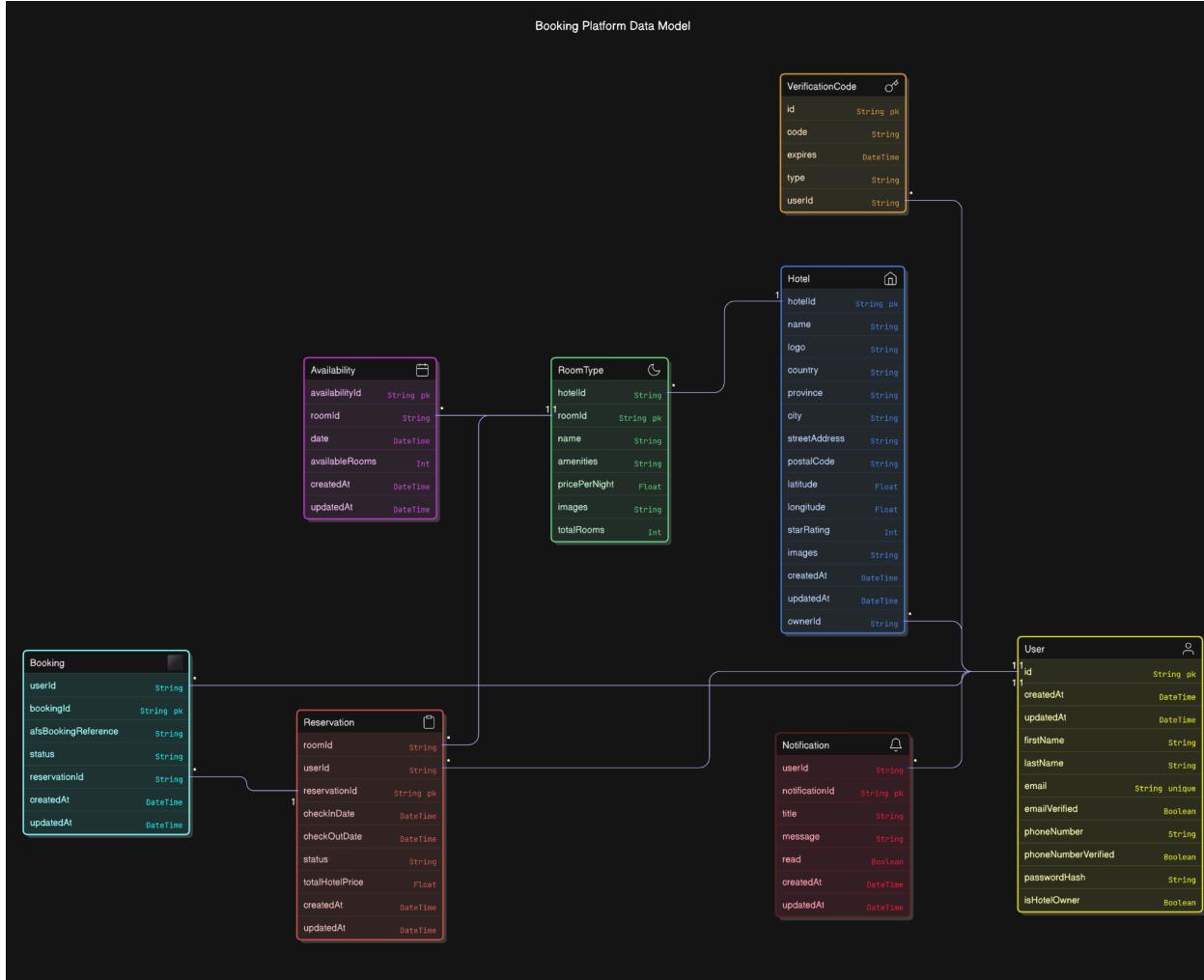


Database Model Design for Hotel Booking System

Entity-Relationship Diagram (ERD)



Database Models and Relationships

1. User Model

Represents system users, including hotel owners and customers.

Fields:

- **id** (Primary Key): Unique identifier.
- **createdAt**, **updatedAt**: Timestamps.
- **firstName**, **lastName**: Personal details.
- **email** (Unique), **emailVerified**: Email details.

- `phoneNumber`, `phoneNumberVerified`: Contact information.
- `passwordHash`: Password storage.
- `isHotelOwner`: Boolean flag for ownership.
- **Relationships:**
 - One-to-Many: Users can own multiple hotels.
 - One-to-Many: Users can have multiple reservations and bookings.
 - One-to-Many: Users can receive multiple notifications.

2. VerificationCode Model

Used for verifying user emails or phone numbers.

Fields:

- `id` (Primary Key)
- `code`: Verification code.
- `expires`: Expiration timestamp.
- `type`: Type of verification (EMAIL/PHONE_NUMBER).
- `userId`: Foreign key to `User`.

3. Hotel Model

Represents hotels in the system.

Fields:

- `hotelId` (Primary Key)
- `ownerId`: Foreign key to `User`.
- `name`, `logo`, `country`, `province`, `city`, `streetAddress`, `postalCode`: Location details.
- `latitude`, `longitude`: Geographical details.
- `starRating`: Default value of 3.
- `images`: Array of hotel images.
- **Relationships:**
 - One-to-Many: A hotel can have multiple room types.
 - One-to-One: A hotel has one owner (User).

4. RoomType Model

Represents different types of rooms within a hotel.

Fields:

- `roomId` (Primary Key)

- `hotelId`: Foreign key to `Hotel`.
- `name`: Room category (e.g., Twin, Deluxe).
- `amenities`: List of available features.
- `pricePerNight`: Price per night.
- `images`: Array of images.
- `totalRooms`: Number of available rooms.
- **Relationships:**
 - One-to-Many: A room type can have multiple reservations.
 - One-to-Many: A room type has multiple availability records.

5. Availability Model

Tracks room availability for specific dates.

Fields:

- `availabilityId` (Primary Key)
- `roomId`: Foreign key to `RoomType`.
- `date`: Availability date.
- `availableRooms`: Number of available rooms.

6. Reservation Model

Represents a customer's hotel reservation.

Fields:

- `reservationId` (Primary Key)
- `roomId`: Foreign key to `RoomType`.
- `userId`: Foreign key to `User`.
- `checkInDate`, `checkOutDate`: Stay duration.
- `status`: Status (e.g., RESERVED, CANCELLED).
- `totalHotelPrice`: Final price of reservation.
- **Relationships:**
 - One-to-One: A reservation can have one booking.
 - Many-to-One: A user can have multiple reservations.

7. Booking Model

Stores booking details, including hotel reservations and flights.

Fields:

- `bookingId` (Primary Key)

- `userId`: Foreign key to `User`.
- `afsBookingReference`: Unique flight booking reference (optional).
- `reservationId`: Foreign key to `Reservation` (optional).
- `status`: Booking status (BOOKED, CANCELLED, PAID).

8. Notification Model

Stores notifications for users.

Fields:

- `notificationId` (Primary Key)
- `userId`: Foreign key to `User`.
- `title`: Notification title.
- `message`: Notification content.
- `read`: Boolean flag for read/unread status.

Relationships Summary

- **User-Hotel**: One-to-Many (A user can own multiple hotels, but a hotel has only one owner).
- **User-Reservation**: One-to-Many (A user can have multiple reservations).
- **User-Booking**: One-to-Many (A user can have multiple bookings).
- **Hotel-RoomType**: One-to-Many (A hotel can have multiple room types).
- **RoomType-Reservation**: One-to-Many (A room type can have multiple reservations).
- **Reservation-Booking**: One-to-One (A reservation can be associated with only one booking).