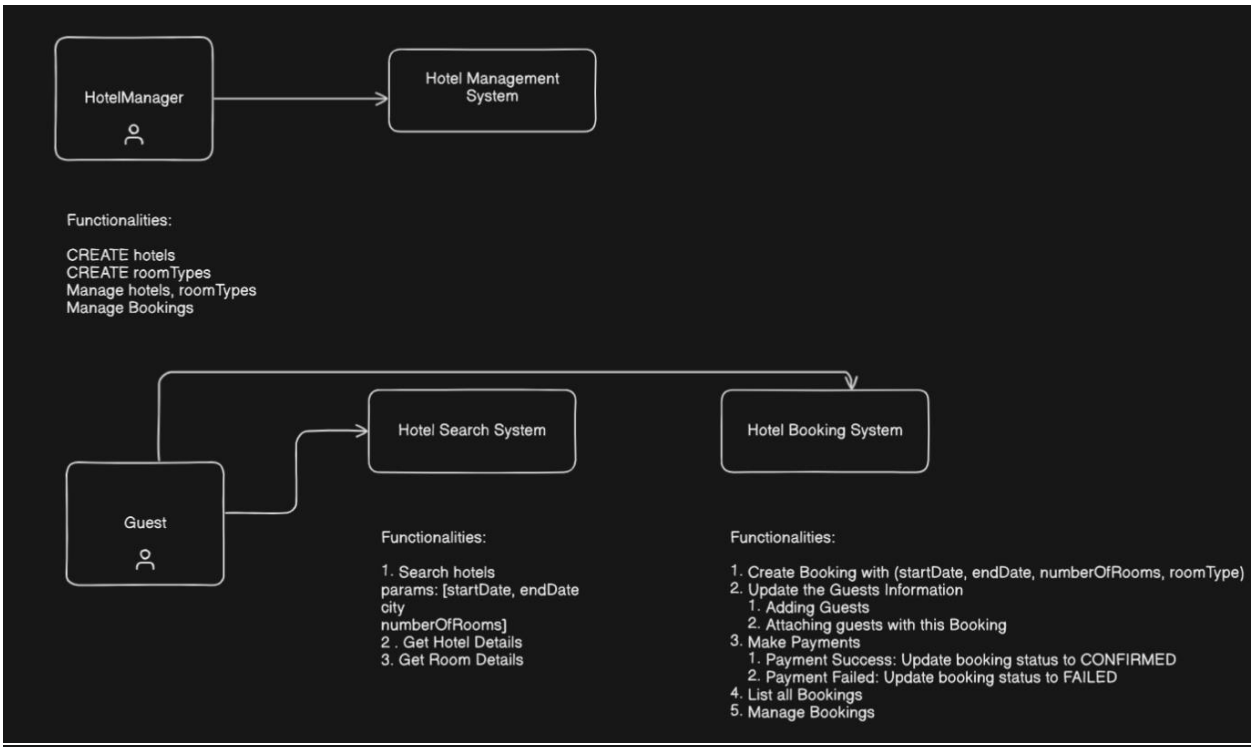
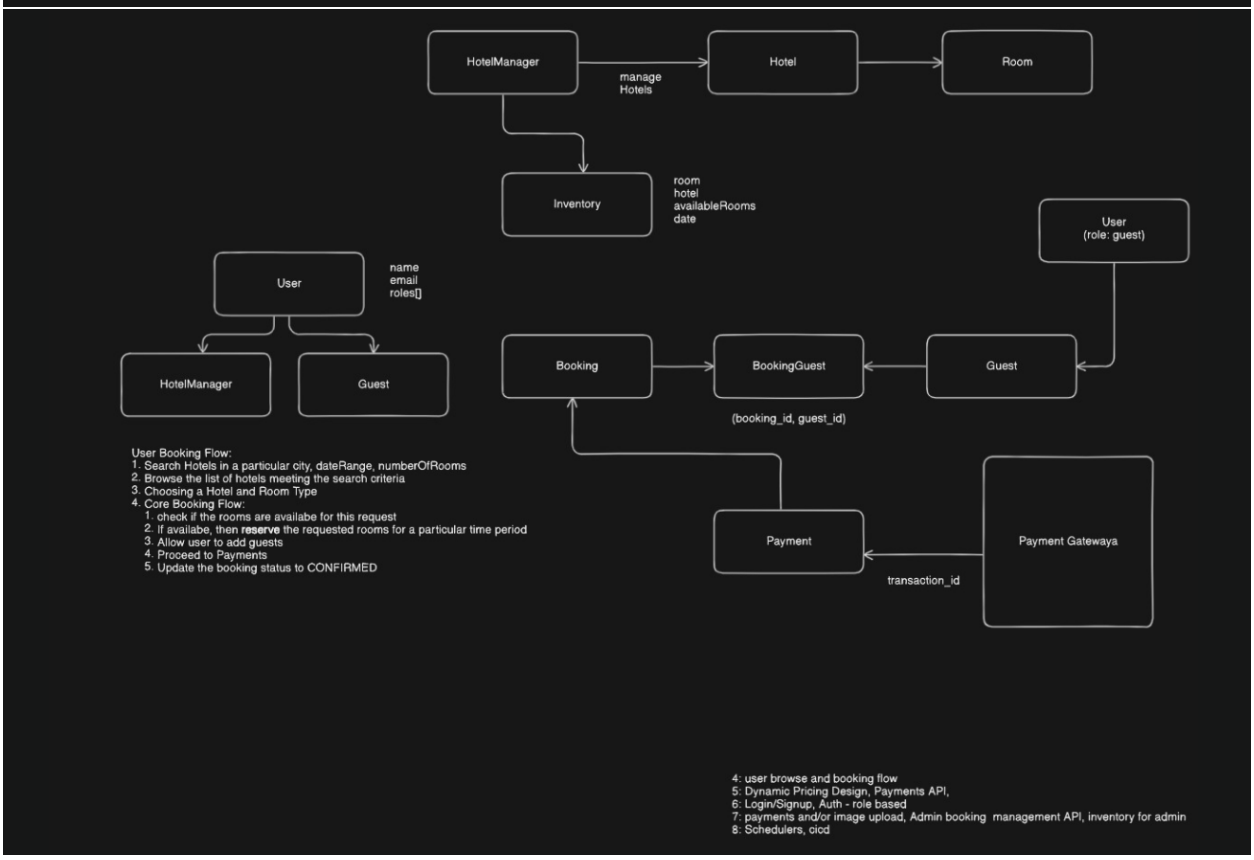

HotelManager
Hotel Management System
Functionalities:
CREATE hotels
CREATE roomTypes
Manage hotels, roomTypes
Manage Bookings
Guest
Hotel Search System
Hotel Booking System
Functionalities:
1. Search hotels
params: [startDate, endDate
city
numberOfRooms]
2 . Get Hotel Details
3. Get Room Details
Functionalities:
1. Create Booking with (startDate, endDate, numberOfRooms, roomType)
2. Update the Guests Information
1. Adding Guests
2. Attaching guests with this Booking
3. Make Payments
1. Payment Success: Update booking status to CONFIRMED
2. Payment Failed: Update booking status to FAILED
4. List all Bookings
5. Manage Bookings


HotelManager
manage Hotels
Hotel
Room
Inventory
room
hotel
availableRooms
date
User
(role: guest)
User
name
email
roles[]
HotelManager
Guest
Booking
BookingGuest
Guest
(booking_id, guest_id)
User Booking Flow:
1. Search Hotels in a particular city, dateRange, numberOfRooms
2. Browse the list of hotels meeting the search criteria
3. Choosing a Hotel and Room Type
4. Core Booking Flow:
1. check if the rooms are availabe for this request
2. If availabe, then reserve the requested rooms for a particular time period
3. Allow user to add guests
4. Proceed to Payments
5. Update the booking status to CONFIRMED
Payment
Payment Gatewaya
transaction_id
4: user browse and booking flow
5: Dynamic Pricing Design, Payments API,
6: Login/Signup, Auth - role based
7: payments and/or image upload, Admin booking management API, inventory for admin
8: Schedulers, ciod

Pages:
Landing Page
Login Page
Search Page
Details Page
Booking Flow Page
  Add Guests (show details and add guests)
  Pay Now (Review page with details and guests)
  Redirect to Bookings page once confirmed.

Dashboard
Profile Page
My Bookings Page (list)
Travellers (list and edit)

Admins:
My Bookings (tabular, paginated)
My Hotels (add and list)
Hotel -> Rooms (add and list)
Inventory -> Calendar view of Inventory. Can update inventory on any date range.

Design Patterns:

Dynamic Pricing Strategy

```
Double getPrice(roomId, startDate, endDate) {

}
```

BasePricingStrategy: The base price
OccupancyPricingStrategy : if booked above 80% then increase price
UrgencyPricingStrategy: If booking within 7 days, then increase price
HolidaysPricingStrategy: Check if it is a holiday, then increae price
DiscountPricingStrategy: If sale is going on, then decrease the price

Strategy Pattern

DecoratorDesign Pattern

APIs

HotelManager:
- POST, GET /api/v1/admin/hotels
- GET, PATCH, DELETE /api/v1/admin/hotels/{hotelId}
- POST, GET /api/v1/admin/hotels/{hotelId}/rooms
- GET, PATCH, DELETE /api/v1/admin/hotels/{hotelId}/rooms/{roomId}

- GET /api/v1/admin/bookings (QP: hotelId, startDate, endDate, status)

- PATCH /api/v1/admins/inventory/{hotelId}/{roomId}/{date}

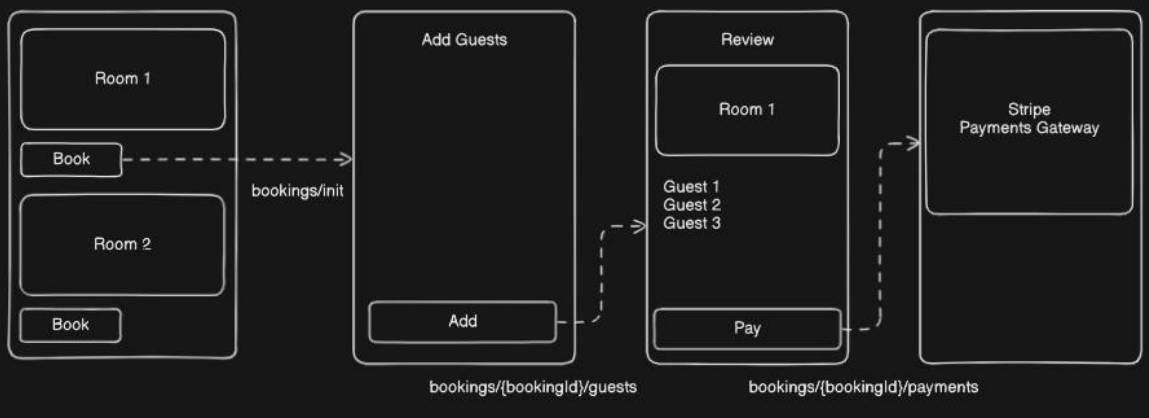
Guest:
- GET: /api/v1/hotels/search (QP: city, checkinDate, checkoutDate, numberOfRooms) : PAGINATED Response
- GET /api/v1/hotels/{hotelId}/getAllRooms (QP: checkinDate, checkoutDate, numberOfRooms)
- GET /api/v1/hotels/{hotelId}/rooms/{roomId}

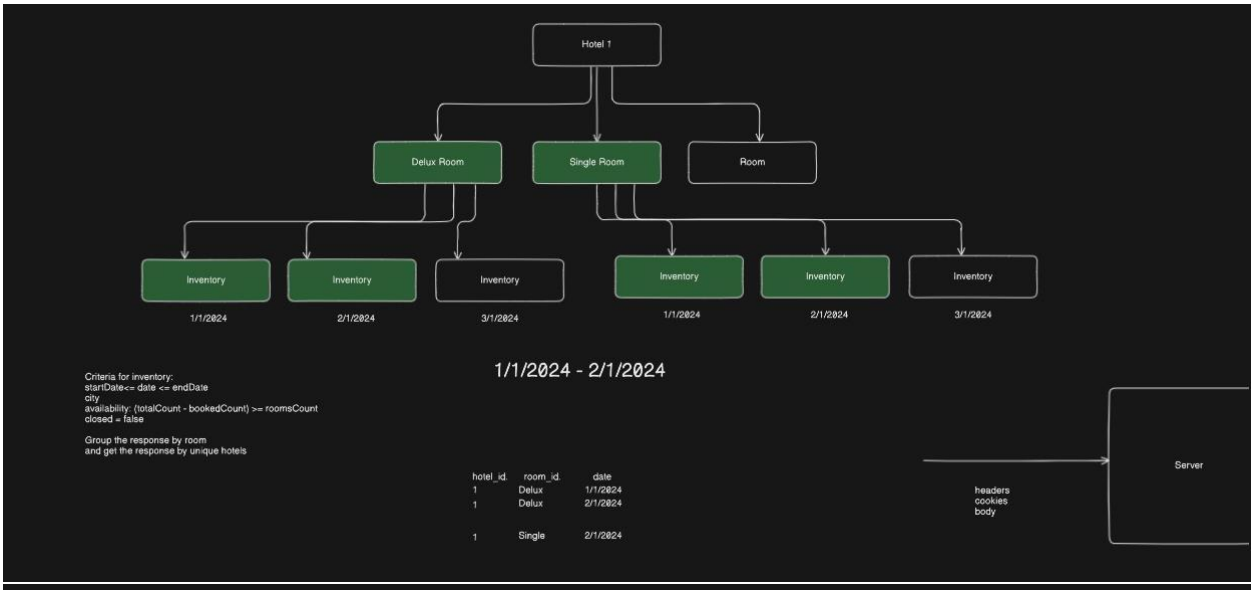
- POST /api/v1/bookings
- POST /api/v1/guests
- PATCH /api/v1/bookings (RB: array of guestId, paymentMethod)
- POST /api/v1/payments/{bookingId}

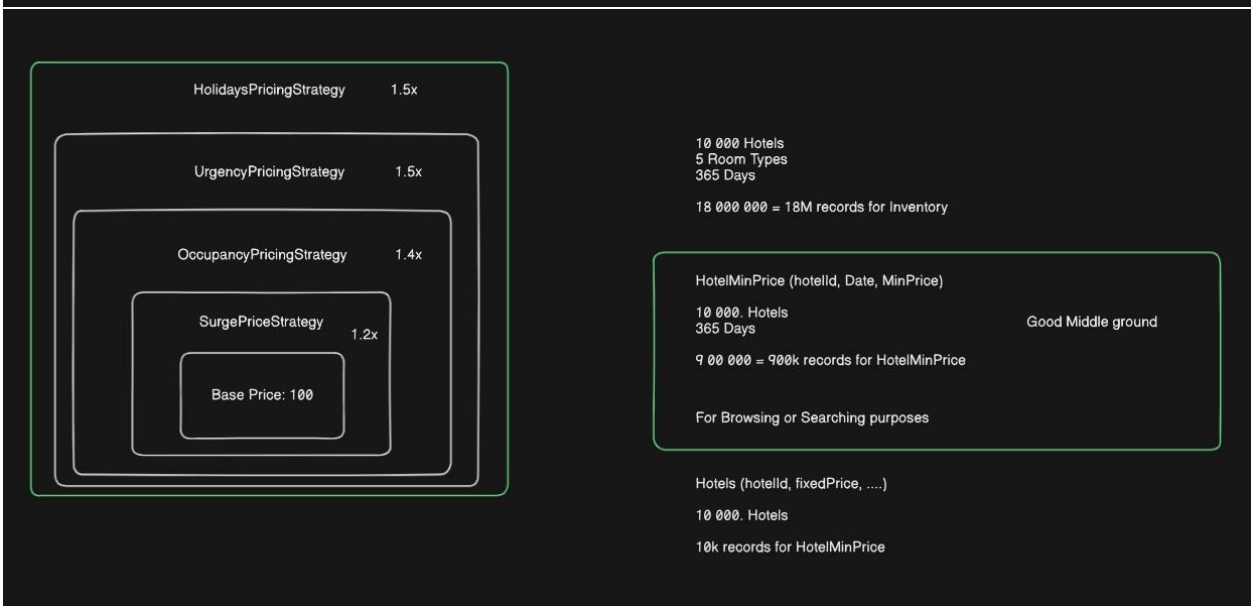
- GET /api/v1/bookings
- GET /api/v1/bookings/{bookingId}
- PATCH /api/v1/bookings/cancel

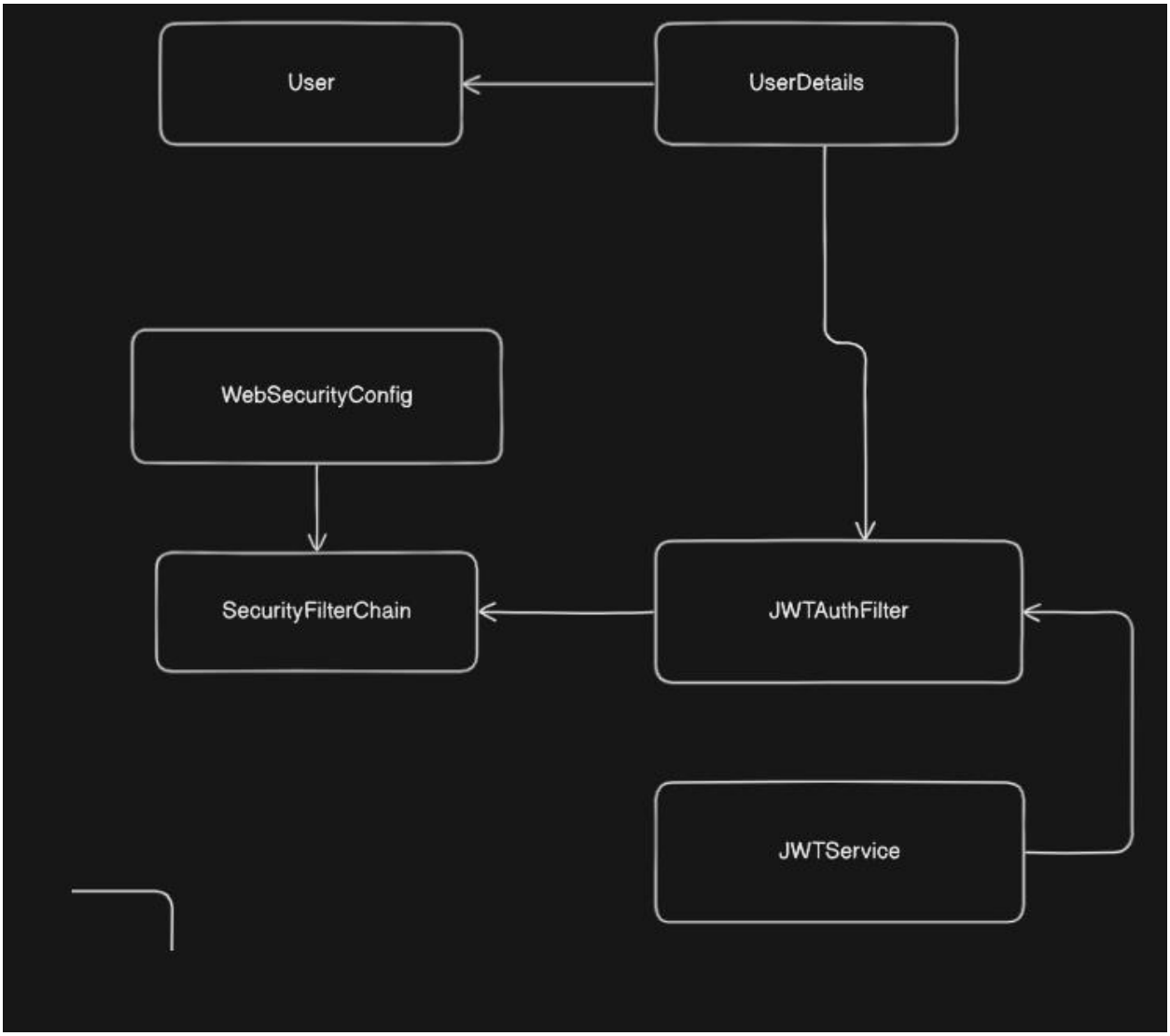
User:
- POST /api/v1/auth/login
- POST /api/v1/auth/signup
- POST /api/v1/auth/verify

System:
- PATCH /api/v1/bookings/resetBookings [Cron Job every 1 minute]




Hotel 1
Delux Room
Single Room
Room
Inventory
Inventory
Inventory
Inventory
Inventory
Inventory
1/1/2024
2/1/2024
3/1/2024
1/1/2024
2/1/2024
3/1/2024
1/1/2024 - 2/1/2024
Criteria for inventory:
startDate<= date <= endDate
city
availability: (totalCount - bookedCount) >= roomsCount
closed = false
Group the response by room
and get the response by unique hotels
hotel_id. room_id. date
1 Delux 1/1/2024
1 Delux 2/1/2024
1 Single 2/1/2024
headers
cookies
body
Server


HolidaysPricingStrategy 1.5x
UrgencyPricingStrategy 1.5x
OccupancyPricingStrategy 1.4x
SurgePriceStrategy 1.2x
Base Price: 100
10 000 Hotels
5 Room Types
365 Days
18 000 000 = 18M records for Inventory
HotelMinPrice (hotelId, Date, MinPrice)
10 000. Hotels
365 Days
Good Middle ground
9 00 000 = 900k records for HotelMinPrice
For Browsing or Searching purposes
Hotels (hotelId, fixedPrice, ...)
10 000. Hotels
10k records for HotelMinPrice


User
UserDetails
WebSecurityConfig
SecurityFilterChain
JWTAuthFilter
JWTService


30 Days
Hotel Rose
switch hotels
Dashboard
Rooms
Bookings
29
Bookings
₹ 2.45 L
Total Revenue
₹ 14,023
Avg Revenue/Booking