Air Cargo Booking & Tracking

Objective

Build a system to create an air-cargo booking and track it through its journey.

Data Model

Booking basic fields

- ref_id (human-friendly, unique)
- origin
- destination
- pieces (int)
- weight_kg (int)
- status (one of: BOOKED, DEPARTED, ARRIVED, DELIVERED)
- timestamps (created/updated)
- Flight ids

Flights

- Flight id
- Flight number
- Airline name
- Departure datetime
- Arrival datetime
- Origin
- destination

Functional requirements

1. **Get Route**
    a. Takes in origin, destination and departure_date.
    b. Return direct flights for given request
    c. Return 1 transit route for a given request.
    Note: Flight in the second hop should be for the same day or next day. Eg DEL-HYD (15th august) + HYD- BLR (16th august). Here, HYD- BLR cannot be after 16th august.

2. **Create Booking**
    a. Creates a new booking using the basic fields above.
    b. Sets initial status to **BOOKED**.

2. **Depart Booking**
    a. Marks a booking as **DEPARTED**.

   b. Records a depart event in the timeline (include where it departed from and optional flight info)..

3. **Arrive Booking**
   a. Marks a booking as **ARRIVED** at a location.
   b. Records an arrive event in the timeline..

5. **Get Booking History (by `ref_id`)**
   a. Returns the booking (basic fields) + full chronological event timeline.
   b. Used by the UI to show tracking.

6. **Cancel Booking:**
   a. Mark a booking as cancelled.
   b. Once a booking has been arrived it cannot be marked as cancelled.

Non Functional Requirements

1. Multiple users might perform different actions on the same booking. So we need to handle concurrency, using distributed locks
2. Considering 50K new booking and 150K updates per day.
3. Number of flights is around 1 lakh and between a given Origin-destination pair there are ~10 flights each day

UI

- Create Booking form.
- Search booking by `ref_id`.
- Booking detail panel showing current status + timeline history.

Logging

- Setup basic logging for future debugging.
- Add logs wherever necessary

Evaluation Criteria

1. DB choice & db modeling
2. Code structure
3. Performance optimization
4. Database indexing
5. Use of caching
6. Unit tests
7. Monitoring
8. UI cleanliness (simple, usable)
9. Documentation (HLD/LLD)

Final Deliverables

1. Backend code.
2. Frontend code.
3. Recorded demo
4. Documentation.