

#programming #systemDesign

Hotel Reservation System

The screenshot displays the Expedia website interface for hotel reservations. At the top, there's a navigation bar with links like 'Home', 'Vacation Packages', 'Hotels', 'Cars', 'Flights', 'Cruises', 'Things to Do', 'DEALS & OFFERS', 'GROUPON Getaways', and 'Rewards'. A 'FILTER RESULTS' sidebar on the left allows users to refine their search by 'Hotel Name Contains', 'Star Rating' (from 1 to 5 stars), 'Price (Avg. per night)' (from less than \$75 to \$300+), 'View Hotels on a Map', and 'Neighborhood' (listing various NYC areas like Manhattan, Chelsea, etc.). The main content area shows search results for 'Hotels in New York (and vicinity) on Apr 10 – Apr 11 for 2 adults'. A 'Hotwire' banner at the top of the results section offers '4-star hotels, 2-star prices' with a 'SHOW ME' button. Below this, a summary table shows average prices: Hotel avg. \$226, 3 star avg. \$193, and 4 star avg. \$254. The results list includes 'Hampton Inn New York Chelsea' (4.4 out of 5 reviews, \$249/night), 'The Belvedere Hotel' (4.3 out of 5 reviews, \$189/night), 'Hilton New York Fashion District' (4.4 out of 5 reviews, \$189/night), and 'DoubleTree Suites by Hilton New York City - Times Square' (4.6 out of 5 reviews, \$249/night). A 'SPONSORED LISTINGS' sidebar on the right promotes 'NYU STERN EXECUTIVE MBA' and '1231 NYC Hotels from \$48'.

High Level Features

- User browses through rooms available for given date range
- User reserves a type of room in a particular hotel
- One check-in, hotel manager assigns a room of that type to the user

Performance Considerations

- When does WRITE happen?
 - User reserves a room
 - User cancels a reservation
 - New hotel or room added

- Browsing through hotel catalog
- Browsing through hotel features

So a significantly higher amount of READ than WRITE

API Requirement

Generic CRUD endpoints for hotel and room management. Let's ignore them.

Reservation

GET /reservations

GET /reservations/123

POST /reservations

DELETE /reservations/123

Data Model

Let's go with a relational database like MySQL or PostgreSQL.

Why?

- Easier to model hotel and reservation data
- More READ than WRITE
- Mostly CRUD operations
- ACID properties, transactional guarantees
- Easier locking mechanisms
- Data can be easily sharded for scalability

Hotel Table

hotel_id	name	address	location
----------	------	---------	----------

Room Table

room_id	room_type_id	hotel_id	is_available
---------	--------------	----------	--------------

Room Type Inventory Table

hotel_id	room_type_id	date	total_inventory	total_reserved

Rate Table

hotel_id	room_type_id	date	rate
----------	--------------	------	------

Reservation Table

reservation_id	hotel_id	room_type_id	start_date	end_date	status	guest_id
----------------	----------	--------------	------------	----------	--------	----------

Guest Table

guest_id	first_name	last_name	age
----------	------------	-----------	-----

What happens when user wants to reserve?

- Check the "Room Type Inventory" table for availability
- If not available:
 - Don't reserve. Throw error.
- If available:
 - Update inventory
 - Create reservation
 - **Both should be done in the same transaction**

How to avoid double booking by the same user?

Let's say the user clicked "Book" twice in very quick succession.
How do we avoid booking twice?

Use an idempotency key

Steps:

- When user lands in final checkout page

- Backend generates a unique key (reservation_id)
- Sends the key to the client
- Client sends the key to the API when reserving
- If user clicks twice:
 - Same key goes to the backend
 - Backend knows a reservation with that key has already been created
 - So Backend throws away the request

Simpler Solution:

- Just gray out and disable the button on client side after being clicked once
- Problem
 - User can disable JavaScript and get around it

How to avoid multiple users reserving the same room?

Approach 1: Use Locking

- Add a new column `version` to the tables
- Client reads the version column when reading a row
- When writing, the application increments the version by 1.
- In the meantime, if version has already been incremented by a different client:
 - Database throws an error
 - Operation is rolled back
 - User will have to try again with a different room

Approach 2: Database Constraint (If supported)

- Add a database constraint
- `CHECK((total_inventory - total_reserved) >= 0)`
- If constraint fails when writing, transaction is rolled back

How to scale?

Database getting too large?

- Nightly batch to remove & archive older rows

- Shard database by `hotel_id`

Read is taking too long?

- Move read traffic from database to cache
- For more popular hotels, cache the inventory information
 - Will lead to more user facing errors
 - Inconsistent inventory data between cache and database
- For all hotels, cache static data like features and hotel details

How can you improve cache data accuracy?

- Database CDC updates Cache
- Whenever inventory changes, cache is invalidated and updated with new inventory