

Microservices Assignment 2023

Business Use Case

Design & Develop a Backend System for Travel Portal as like MMT or Booking.com, The business idea is to bridge the gap between service providers (Suppliers – Ex- indigo, Lemon Tree ... etc) and service receivers (Consumers - Users).

- Service Receivers (Consumers) can book Flight Tickets & Hotels. They can avail all these services via placing an booking through the Mobile App / Portal.
 - Create a Login for the application and after successful login, User can do the Flight Search & Hotel Search.
 - If the type is Flight, Search should have all input parameters as Departure Location, Arrival Location, Flight Class (Economy/Business)., Flight Date.
 - If the type is Hotel, Search should have all input parameters as Hotel Name, City, Check In Date, Check Out Date Output.
 - Once Search is triggered after specifying the search parameters, It will display the search results.
 - User can also shortlist the Hotels only.
- Next Step, Consumer will create a Booking.
- When a service is requested, Booking is generated at the backend system and then various background validation happens in the system with respect to availability & payment confirmation... so on and so forth. Based on validation status, booking status will be updated in the system.
- Consumer will also get a notification with respect to the booking confirmation and service provider details & Service Provider will also get the Detailed Description for customer & Other details (ex- Address, Contact Number of Consumer).
- In case of Validation Failure, Consumer will receive notification for Booking cancelled, At the same time, System must take care for refund process.

Tools/Technologies

1. Any language (Java/.Net/Python/NodeJS) of choice for writing microservices
2. Service Discovery (Eureka for Java/DotNet, other languages can choose any)
3. API Gateway for implementing routing (Zuul/SpringCloud for Java, Ocelot for DotNet, other languages can choose any)
4. Cross-cutting concerns to be taken care of
 - Circuit-Breaker Pattern (Fail-over mechanism)
 - Distributed Tracing & Logging

Deliverables

1. A writeup of identified microservices for above use case along with reasonable explanation (with High Level Diagram).
2. URL definitions of the scenarios (API Endpoints -Write up for Expose API Endpoints for each Microservice with request/response payload)
3. API Gateway, Service Discovery Integration
4. Source code of all microservices.
5. Write up for Inter-communications approach & assumptions -Which contain relevant assumptions if taken and how communication is happening between your microservices.
6. Docker File should be incorporate for each microservice. Docker images to be built using Docker file and pushed to docker hub. (Please share the image path for the same for each microservice in document)
7. A docker-compose file for booting the applications from a single file. Intention is that your application should be able to get up and running on any system.

Guidelines:

1. This assignment requires to identify all the microservices for above mentioned use case and expose the API endpoints.
2. **No User interface is required, you need to design backend system only.**
3. Integration with database is not mandatory, in memory data structures (lists, maps) can be used to store data temporarily for the Hotels & Flights (Create some dummy data for Inventory).
4. Identify & implement proper inter-service communication (Sync/Async).
5. Mention relevant assumptions taken while implementing application (**one successful booking flow and one failure booking flow with refund process are must from the implementation point of view**).

Assumptions Like:

- For Notification, no external integration required, Logs will work for us.
 - User can't book the flight & hotel in single booking, they must perform separate request for each, but they can book multiple rooms in a hotel for selected dates & similarly flights as well (Ex- Return Trip Flight -> Two way)
 - Misleading information / outdated information etc. (Ex- Prices are highly fluctuating in this industry, you can take an assumption for static prices)
 - Overlapping issues (i.e., seat being taken when you are in the process of payment. This is case of inventory out of stock at run time, you have to take care of failure scenario.)
 - Authentication is mandatory (Can we mocked using In-memory Database/Collections), but you can ignore the authorization process.
 - Soon...
6. Your solution should be able to build/compile and run.
 7. Standard coding guideline should be strictly followed.

Priority wise parameters on which we will be evaluating the assignment:

- Write-up for Microservices Identification with HLD
- Write up for Expose API Endpoints for each Microservice (with request/response)
- Implementation of API Gateway & Service Discovery (Every request will be landed to API gateway & every microservice should register with service discovery)
- Implementation for all microservices with respect to above use case (one Positive Flow & one Negative scenario) & Interservice Communication Approach.
- Cross Cutting Concerns
 - Logging & Circuit-Breaker Pattern is mandatory to implement.
 - Distributed Tracing is optional. (No marks will be deducted)
- Docker File
- Docker Compose to deploy the solution.

Assignment Submission: 2 weeks from the Start Date.