Movie Booking Platform

High Level Design Document

XYZ wants to build an online movie ticket booking platform that caters to both B2B (theatre partners) and B2C (end customers) clients.

Key goals it wants accomplished as part of its solution:

* Enable theatre partners to on board their theatres over this platform and get access to a bigger customer base while going digital.
* Enable end customers to browse the platform to get access to movies across different cities, languages, and genres, as well as book tickets in advance with a seamless experience.

Theatre Service:

1. ***https:/{host}/movie-booking-platform/theatres* Type – Post**

Payload – Theatre details

1. ***https:/{host}/movie-booking-platform/theatres* Type – Get**
2. ***https:/{host}/movie-booking-platform/theatres/{id}* Type – Get**
3. ***https:/{host}/movie-booking-platform/theatres/{id}***  **Type – Put**

Same payload as above will be used for theatre maintenance

Search Service:

1. ***https:/{host}/*** ***movie-booking-platform/search/{theatreId}* Type – Get**
2. ***https:/{host}/*** ***movie-booking-platform/search* Type – Get**

This service will be used to fetch with multiple search criteria

Booking Service:

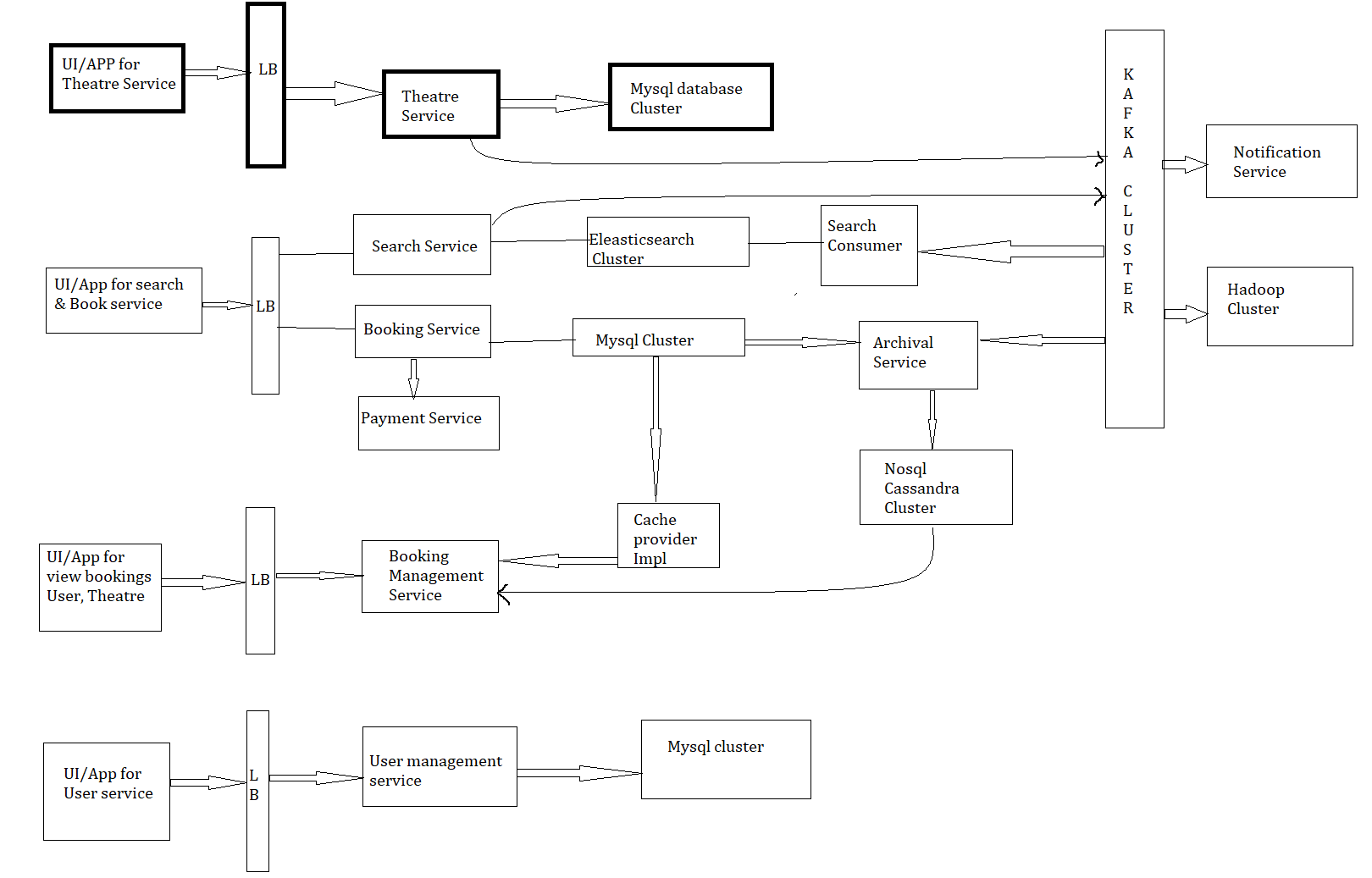
1. ***https:/{host}/*** ***movie-booking-platform/book* Type – Post**

Payload – booking payload containing movie name, theatre and number of seats

Tech Stack:

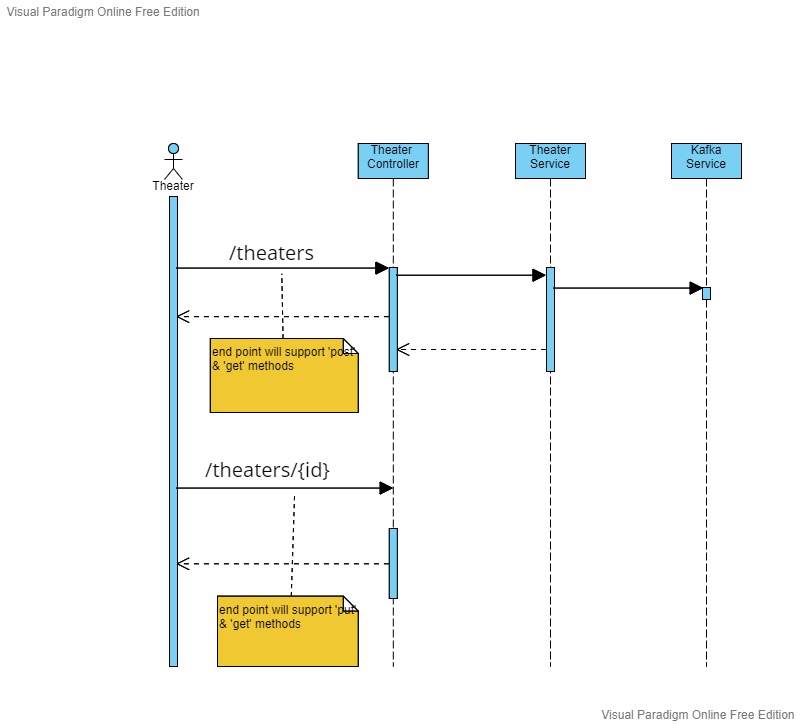
* Java with SpringBoot, docker, Kubernetese, Cucumber framework
* Database – any RDBMS
* Hosting environment - Any public cloud
* Testing strategy – Unit testing, automation testing (BDD)
* Logging – Splunk or Scalyr (Alerts and dashboards will be used for reporting)
* CI/CD – Git, Jenkins
* Security – At API gateway level (JWT token based)
* Disaster recovery & Auto scaling – Deploy application in more than two regions with 3 self-healing pods. Configure auto scaling if CPU consumed by more than 75%.
* Make automation suite part of Jenkins pipeline and fail the build if automation fails
* Integrate with quality gate tools like Checkmarx, Blackduck to resolve the OWSAP top 10 vulnerabilities

**System Design**



**Sequence Diagram:**

Representing flow for only one service.

****

**Note** – Due to time constraint I haven’t added API granular details here. Same can be discussed with the help of the code.