

# A Survey on Seat Reservation System Using Microservices Architecture

Kulin Galundia\*, Poornima Kulkarni\*\*

\*Department of Information Science ,R.V. College of Engineering,Bangalore (Karnataka),560059

Email: [kulingalundia.is17@rvce.edu.in](mailto:kulingalundia.is17@rvce.edu.in)

\*\* Department of Information Science ,R.V. College of Engineering,Bangalore (Karnataka),560059

Email: [poornimapk@rvce.edu.in](mailto:poornimapk@rvce.edu.in)

\*\*\*\*\*

## Abstract:

This Paper describes an office seat booking system that allows an organization to enable the employees a facility to pre book a seat in the form of a reservation when he/she desires to visit the office in this work from home era. Organizations can save a lot of money on infrastructure thanks to the seamless adoption of shared spaces and increased space automation. When social distancing is required, seating arrangements can be planned by filling only alternative desk spaces, as well as restricting the number of individuals in a given area, both of which can be controlled only with the use of a competent space management system. By tracking the space utilisation across all areas, facility managers can save costs by altering the operational parameters like HVAC & lightening operational hours , fresh-air intake , set temperatures etc. This will also result in equipment maintenance and replacement costs.

**Keywords** —Reservations, Team seats, Floater seat, Social distancing, Spring Boot, MongoDB, REST API, Elastic Search , Feign Client, Jenkins, Swagger, React JS

\*\*\*\*\*

## I. INTRODUCTION

In this expanded work-from-home era, the Seat Management System is a workplace space planning solution that will allow us to efficiently track and manage team-based space allocation, seat assignment, and on-demand seat booking of shared workspaces in the form of reservations. If an employee of a workplace wants to visit the office then he/she can book a seat for an entire day based on their teamId. One can also book a floater seat if he/she is not assigned a teamId or for any other purpose.

The COVID-19 outbreak and social distance rules have had an impact on office arrangements, and the Seat management system is designed to assist us manage space efficiently while adhering to social distancing rules. It will enable us to arrange just alternate workstation space allocation in order to maintain “physical” or “social” separation norms.

The system would have different user modules including employee , facility manager or administrator each with some specific permissions and roles. The employees can book a reservation for a maximum of next 5 days providing certain details like reservation date, hasMeal, type of seat (Team

seat or Floater Seat) followed by selecting the seat. The facility manager will have certain additional permissions as compared to the employees and will handle facility requests. The administrator can add a seat, or can mark a seat as available or disabled for any particular use case.

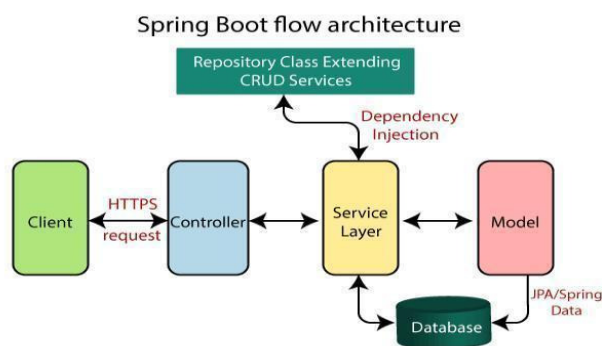
Based on the duration/timing, the system may intelligently fill spaces area-by-area/floor-by-floor in the most efficient manner. The system frees the seat if an employee does not come within a specific freeze time so that particular seat can be assigned to another employee. Also if an employee wants to book a Team seat and finds all seats to be booked then he/she can make a facility request to the facility manager for a floater seat.

## 2. DIFFERENT SOLUTIONS, TOOLS AND CONCEPTS USED

This section examines a variety of technology-based options for the usage of seat booking systems and shows how Spring Boot and MongoDB may make a significant difference:

1. In the paper “Using MongoDB and Spring Boot to Create a RESTful Web Service”, a paper that describes how Spring Boot facilitates the bootstrapping and development of new Spring

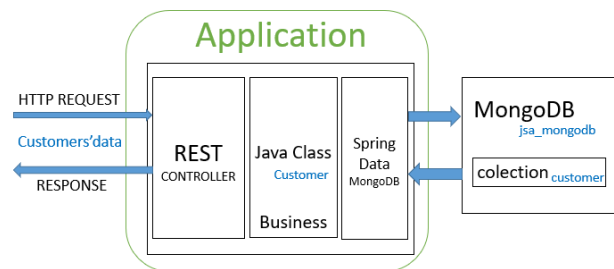
applications, according to this study. The framework provides an opinionated approach to configuration, removing the need for developers to write boilerplate code. MongoDB is a document database that is simple to use and set up. An application programme interface (API) that employs HTTP requests to GET, PUT, POST, and DELETE data is known as a RESTful API. We employ a microservices architecture to accomplish our goals. It's used to create and deploy services on its own. Each running service has its own process, resulting in a lightweight paradigm for supporting business applications.



2. This paper discusses how microservice SpringBoot can be used to design & develop a Backend Application for Public Complaint Systems. It illustrates how to create a public complaint service application using the spring boot microservice architecture and a web application. The microservice architecture is used to divide the app functionality into different divisions, or many microservices, depending on business processes, which were then integrated to form a complete business process single app. One of the advantages of this design is that it allows for the addition of more microservices without causing any problem to others. The application can be accessed through a web browser due to hosting in the cloud. The e-government systems can take advantage of the

usage of microservice technology. In microservices, the core concept of modularity allows us to manage the services that lie in some other application. The impact of a certain service's development is that it will not come in the way with other services. A service's capacity building is often differentiated from other services in order to ensure that the resource is properly utilised. In addition, numerous programming languages can be used to construct services.

3. REST API: Java Spring Boot and MongoDB . Thomas Gleason, describes how SpringBoot framework and MongoDB are used to create a RESTful web service. The Java Spring Boot framework provides a strong set of tools for front-end and back-end web development. It offers built-in security and database access setup, as well as easy request mappings. This framework is very flexible and simplifies web development.



4. Spring Boot Reference Documentations, provides a brief overview of the documentation for spring boot. The documentation includes installing spring boot, dependency management, spring boot features , spring boot actuators, deploying spring boot applications, and spring boot CLI along with build tool plugins. Within the Java ecosystem, there are 2 stable, mature stacks for building web applications and Spring framework becomes a proper solution as it implies recognition and powerful adoption. Spring provides a quite powerful way in creating a web application due to the support for dependency injection. It also provides support for transaction management and persistence of polyglot. It provides some additional features like first-hand support for REST API , a framework (MVC), application security, and a variety of more features. The features of Spring boot would

include starter modules for simplification of configuration of dependencies. It also provides embedded builtin Tomcat, Undertow etc. Whenever possible, it assists in auto configuration. It is surely responsible for building stand-alone Spring apps. The other features would include:

- Metrics, health checks, and externalised configuration are some examples of production-ready features.
- no requirement for XML configuration

5. MongoDB Reference Documentations, describes mongoDB as a database for documents. It is designed in such a way that development would be easy along with scaling. The Documentation explains some powerful concepts in MongoDB, presents the command language, and is highly responsible for yielding considerations and procedures that are operational and administrative, also as a reference section comprehensively. It includes the guide on mongo Shell, CRUD operations, aggregation, transactions, indexes, replication, sharding, storage etc. The features of MongoDB includes:

- Ad-hoc searches for real-time analytics that are optimised.
- Appropriate indexing for faster query execution.
- For enhanced data availability and stability, replication is used.
- Sharding.
- Load Balancing

6. ReactJS Reference Documentations, provides a brief overview of the documentation on ReactJS. It describes React as a JavaScript library for building user interfaces.

It includes the key concepts and API references. It also includes Hooks, Testing, Concurrent mode etc. The features of React includes:

- JSX is a JavaScript syntactic extension.
- The “virtual” DOM is the lightweight approximation of the “real” DOM that React retains in memory (VDOM)
- One-way Data Binding

- Debugging
- Performance
- Extensions

7. Survey Paper on ElasticSearch, describes ElasticSearch as a means to organise data and make it conveniently accessible, according to the author. It's a Lucene-based server-side search engine. It's a full-text, highly scalable, distributed software. Java is used to create elastic search. It's open source and released under the Apache License. The most widely used enterprise search engine is Elasticsearch. Elasticsearch incorporates all security, scalability, hardware efficiency, and speed improvements. It is a technique for querying written text. It can do other clever things, but its main function is to return text that looks almost exactly like a query and to run statistical studies on a large amount of text. It is a Java-based database that could be considered as a stand-alone server that uses the HTTP/JSON protocol to collect data and optimise it for searches based on languages before storing it in a complex manner. ES is particularly user-friendly, as it comes with clustering and leader selection built-in. Elasticsearch is fantastic with respect to crawled web pages for finding similar text in a body, whether it's searching a database of trade related products by detailed description. Elasticsearch is a fantastic programme.

8. This paper discusses how Jenkins evolved to a Continuous Delivery platform from a pure Continuous Integration platform. It also describes how Jenkins helps to deploy your services acting as a free automation server. As a result it embraces assisting in software development relating to building, deploying and testing. Jenkins becomes a built-in java orchestrator tool for providing us with a number of plugins so that virtually we can have support for automation. Jenkins provides analysis of static code and running tests to detect bugs. It runs servlet containers like Apache tomcat. The purpose of this article is not just presenting jenkins as a CD hub, but also to highlight the problems that must be overcome so that tracking abilities of jenkins can be improved.

9. This paper discusses the importance of usage examples in REST API documentations. It includes describing development and updation of REST API documentation which includes usage examples. It describes how Usage examples can be time-consuming and result in costly operation for evolving apis. Most REST API documentation solutions mainly aim on how description of API objects can be automated, but also how recording usage examples requires considerable effort. As a result, while prioritising documentation efforts, REST API developers should weigh the cost vs. benefit of incorporating usage examples in the documentation. In order to better understand the challenges that REST API client developers face while using an API that lacks usage examples, a controlled study was undertaken with some well qualified software engineers. REST API client developers face productivity challenges when using the required data types, HTTP headers , request body and data formats when documentation lacks usage examples. By studying and analysing the REST API documentation suggestions in this article, REST API developers can reduce errors, raise API client developer satisfaction and improve success rates.

10. Rest API testing with Spring Cloud Feign Clients, a blog that describes Spring cloud feign which is a convenient way to test your application's API, focused on creating tests to uphold business logic instead of spending time on the technical implementation of web services clients. The only thing that needs to be described is how to reach the remote API service by providing details such as the request, URL and response body, accepted headers, etc. The Feign Client will be taking care of the implementation details. In this blog post there is a step by step guide describing the process of creating a comprehensive framework for testing RESTful APIs. Spring Feign cofunction with other Spring Framework components like Spring Web and Spring IoC container instead of standing alone . A closer look to these two components is required to understand feign client.

11. Swagger documentation describes how Swagger enables us to explain the structure of our APIs in a machine-readable format. The capacity of APIs to define their own structure is at the heart of Swagger's greatness. What makes it so special? It can automatically generate beautiful and interactive API documentation by reading the structure of our API. We can produce client libraries for our API in a variety of languages automatically, and we can also investigate other options such as automated testing. Swagger accomplishes this by requesting a YAML or JSON response from our API that offers a descriptive detail of the whole API. This file contains a resource list for our API, which follows the OpenAPI Specification.

12. Sumo Logic documentation describes Sumo Logic as a platform that allows us to make data-driven decisions and save time investigating security and operational concerns, allowing us to focus on more critical tasks. We can improve the efficiency of our troubleshooting. With cloud-based contemporary analytics powered by machine learning, Sumo Logic helps us cut downtime and shift from reactive to proactive monitoring. Sumo Logic gives real-time visibility into cloud apps and infrastructure on AWS, Azure, and GCP.

### III. CONCLUSION

The Project overall proves the need for a Seat Booking System across organisations. It emphasises how implementing seamless shared spaces and improving space automation may help businesses save a lot of money on infrastructure. It also provides the different concepts used for building SMS along with an architectural roadmap. This paper discusses the techniques and concepts used for building the application. A brief of each of the concepts used has been mentioned . It clearly describes which concepts could be used for which particular use cases. The paper gives an overall project structure idea along with the implementation's perspective.

### REFERENCES

[1] Robert Rice, “Using MongoDB and Spring Boot to Create a RESTful Web Service”, November 2018

[2] Hatma Suryotrisongko, Dedy Puji Jayanto, Aris Tjahyanto, “Design and Development of Backend Application for Public Complaint Systems Using Microservice Spring Boot”, 2017 , Procedia computer science

[3] Thomas Gleason, “REST API: Java Spring Boot and MongoDB” , May 2018

[4] Spring Boot, "Spring Boot reference documentation" [Online]. Available: <https://docs.spring.io/spring-boot/>

[5] MongoDB, “MongoDB reference documentation” [Online]. Available: <https://docs.mongodb.com>

[6] ReactJS, “ReactJS reference documentation”, [Online]. Available: <https://reactjs.org/docs/>

[7] Pragya Gupta, Sreeja Nair , “Survey Paper on ElasticSearch”, 2013 , International Journal of Science and Research (IJSR)

[8] Valentina Armenise, “Continuous Delivery with Jenkins: Jenkins Solutions to Implement Continuous Delivery”, May 2015,

[9] S M Sohan, Frank Maurer, Craig Anslow, Martin P. Robillard, “A Study of the Effectiveness of Usage Examples in REST API Documentation” , 2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)

[10] Dzmitry Ihnatsyeyu , “Rest API testing with Spring Cloud Feign Clients”, April 2018, Blazemeter

[11] Swagger, "Getting Started With Swagger," [Online]. Available: <https://swagger.io/getting-started/>.

[12] Sumo Logic , “Sumo Logic reference documentation”, [Online]. Available: <https://help.sumologic.com/01Start-Here/Quick-Start-Tutorials>