

Restaurant Services

Spring Boot 301 Project Case Study

Version 1.0 | 10-Jul-2019

Copyright Information

This document is the exclusive property of Mindtree Limited (Mindtree); the recipient agrees that they may not copy, transmit, use or disclose the confidential and proprietary information in this document by any means without the expressed and written consent of Mindtree. By accepting a copy, the recipient agrees to adhere to these conditions to the confidentiality of Mindtree's practices and procedures; and to use these documents solely for responding to Mindtree's operations methodology.

Revision History

Please keep the latest version on top

Ver	Change Description	Sections	Date	Author	Reviewer	Approver
1.0	Initial Draft		10-JUL-19	KAVIARASU VENU		KRISHNAN K S

Table of Contents

1. Overview	4
2. Features (Functional Requirements)	4
2.1 Restaurant Search Service.....	4
2.2 Order Management Service	4
2.3 Customer Management Service	5
2.4 Review Management Service.....	5
3. Non-Functional Requirements	5
3.1 Security.....	5
3.2 Performance	5
3.3 Scalability.....	5
4. Advanced Features	6
5. Deliverables	6
6. Acceptance Criteria for standard set of features and NFRs	7
7. Acceptance Criteria for Advanced features.....	8
8. Reference Implementation:	8
Mindtree Limited	9

1. Overview

A global hospitality company looking for a team who can help them in building rapid APIs to access their chain of restaurants and manage orders. APIs should be deployed and hosted independently based on following service category. This should help their customers to use on need basis:

- Restaurant Search Service
- Order Management Service
- Customer Management Service
- Review Management Service

2. Features (Functional Requirements)

2.1 Restaurant Search Service

In this service, customer can search for the restaurants based on multiple options:

- Based on location
- Distance
- Cuisine
- Budget
- Ratings
- Restaurant name
- Menu of restaurant

2.2 Order Management Service

This service should help customer to take care of:

- Placing order
- Updating order
- Cancelling order
- View order
- Calculating the total amount based on the number of items ordered.

2.3 Customer Management Service

This service provides features like:

- Registration of the customer
- Updating customer details
- Deactivate Customers

2.4 Review Management Service

This service should help customers:

- To view and provide reviews and ratings for the restaurant
- Customer can also update their reviews and ratings.

3. Non-Functional Requirements

3.1 Security

- The customer management service must be secured with authentication either using social login/custom JWT based authentication

3.2 Performance

- The search service should be completed within 300 Milliseconds
- Ordering of food should be completed in 500 Milliseconds

3.3 Scalability

- The application must support for 10000+ restaurants with at least 2000 users

3.4 TDD

- Implement test cases before implementing features
- There should be minimum 90% of code coverage

3.5 Logging

- Monitor how customers are using APIs to check unpredictable scenarios and unintentional misuse

3.6 Pagination

- The search result should support pagination

4. Advanced Features

Learners can pick any of below advanced features and implement end to end. Acceptance criteria would vary based on track selected.

Microservices with AWS	Customer Sentiments and Brand Value
<p>Use an existing application codebase to migrate to AWS to create cloud native application.</p> <p>Use following major services:</p> <ol style="list-style-type: none"> 1. Amazon Lambda 2. Amazon Elastic Container Service 3. Amazon S3 4. Kubernetes 	<p>Create user analytics service which focuses on analyzing user reviews and brand value of restaurants:</p> <ol style="list-style-type: none"> 1. Simulate customers feedback 2. Cleanse the data 3. Create APIs which provides some of the interesting data on user reviews 4. Create a visual dashboard with various trends
<p>Code Reference: :</p> <p>https://github.com/zhouyumeng1077/Spring-Boot-Microservices-Based-Online-Ordering-System-</p>	<p>Reference: https://medium.com/cindicator/building-microservice-for-twitter-analysis-weekend-of-a-data-scientist-d37074956368</p>

5. Deliverables

API documentation with swagger URLs for each services
Screen capture of code coverage report
Jmeter performance report with mocking of 2000 users
Log analysis report on customer activities

6. Acceptance Criteria for standard set of features and NFRs

The final release must support following functional and non-functional requirements:

6.1 Functional Requirements:

Feature List
Search Service : Allowing customers to search based on at least 3 to 4 parameters
Order Management Service: Allowing customers to book and manage orders made – placing, canceling and viewing orders
Customer Management Service: Allowing customers to manage user base – Register, Update and deactivate users
Review Management Service: To view, provide and update ratings

6.2 Non-Functional requirements

Feature List
Authentication: Using JWT token based to access Order Management Service
Performance: <ul style="list-style-type: none">• The search service should be completed within 300 Milliseconds• ordering of food should be completed in 500 Milliseconds
Scalability: The application must have at least 10000 Restaurant details and 2000 users in the database
Code Coverage: The final release must have code coverage report of 90%
Log Report: Contextual report analysis with insightful details – Average of number of users accessed per day, User base based on regions etc.,

7. Acceptance Criteria for Advanced features

Microservices with AWS	Customer Sentiments and Brand Value
<ul style="list-style-type: none">• API documentations using Swagger• Creation of at least 3 services using AWS Lambda• Deployment of services using container services• Managing services using Kubernetes	<ul style="list-style-type: none">• API documentations using Swagger• At least 5 interesting APIs which reflects user reviews and sentiments:<ul style="list-style-type: none">○ Region wise Top rated restaurants○ Positive words○ Negative words <p>Creation of trends using visual dashboard</p>

8. Reference Implementation:

1. <https://developers.zomato.com/documentation>
2. <https://medium.com/vandium-software/5-easy-steps-to-understanding-json-web-tokens-jwt-1164c0adfcec>
3. <https://rapidapi.com/restaurantmenus/api/us-restaurant-menus>



Mindtree Limited

Contact Information

<Specify Contact Name>

Email: <Specify Contact E-mail address>

Global Village, RVCE Post, Mysore Road,
Bangalore-560 059

INDIA

Phone- 91 (80) 6706 4000

Fax- 91 (80) 6706 4100

Web- www.Mindtree.com