High-Level-Design Document

Team: Tech Wizards

CMPE 202 Software Systems Engineering

1. Introduction

The RestaurantFinder web application allows users to search for restaurants, submit reviews, and view ratings. It supports three roles: Users, Business Owners, and Admins. The system is designed to be deployed on Azure Cloud with auto-scaling and load balancing, using React for the frontend, Spring Boot for the backend, and MySQL for the database.

2. System Architecture

The architecture follows a three-tier structure:

Frontend: Developed using React for responsive UI.

API Layer (Backend): Built with Spring Boot to handle all business logic and RESTful communication in JSON.

Database: MySQL, hosted on Azure Database for MySQL.

3. Components

Frontend (React):

Provides role-based user interfaces for Users, Business Owners, and Admins.

Connects to the backend through APIs for features like search, reviews, and user management.

Backend (Spring Boot):

Handles core functionalities such as search, restaurant management, review submission, and admin features (e.g., managing duplicates).

Role-based access control through JWT authentication.

Error handling and input validation are critical for securing APIs.

Database (MySQL):

Tables:

Users: Stores user details and roles.

Restaurants: Holds restaurant information, including name, location, cuisine, and ratings.

Reviews: Contains user reviews and ratings for each restaurant.

Business Owners: Connects business owners to their respective restaurants.

4. Deployment on Azure Cloud

Azure App Services: Hosts the frontend and backend for scalability.

Azure Load Balancer: Ensures high availability and distributes traffic evenly.

Azure Auto Scaling: Automatically adjusts the number of instances based on demand.

MySQL: Hosted using Azure Database for MySQL, ensuring data integrity and scalability.

MapBox API: Integrated for location-based restaurant searches by zip code.

5. Key Functionalities

For Users:

Search for restaurants by name, cuisine, price, rating, or location.

Submit reviews and ratings.

View restaurant details, reviews, and ratings.

For Business Owners:

Add and manage restaurant listings (name, address, hours, etc.).

Upload photos and update restaurant details.

For Admins:

Remove duplicate restaurant entries.

Remove closed restaurant listings from the system.

6. Security

JWT tokens for secure user authentication and role-based access.

HTTPS for secure communication between clients and servers.

7. Scalability Considerations

Azure Auto Scaling ensures the system can handle increased traffic by dynamically adjusting resources.

The database (MySQL) is designed to handle high traffic and large datasets with optimized indexing and query handling.

9. Project Management

GitHub Repository: The codebase, weekly reports, and project artifacts are maintained on GitHub.

Scrum Process: Scrum backlog, sprint tracking, and burndown charts are maintained to track progress.

XP Core Values: Emphasis on communication and simplicity during the project.

10. Conclusion

The RestaurantFinder application is designed to scale efficiently on Azure Cloud, offering a robust and secure platform for users, business owners, and admins. The system architecture ensures high availability, security, and performance, making it a reliable solution for restaurant management and discovery.