Software Requirements Specification

for

Multi Restaurant using React and Spring boot

Prepared by Sripriya(211801340013) Aryaman(211801340020)BhanuSriteja(211801340002)Mahesh(211801340019)Suresh(211801340016)Indu(211801340005)

Centurion University of technology and management

Table of Contents

Table of Contents ii

1. Introduction 3

1.1 Purpose 3

1.2 Document Conventions 3

1.3 Intended Audience and Reading Suggestions 4

1.4 Project Scope 5

1.5 References 6

2. Overall Description 6

2.1 Product Perspective 6

2.2 Product Features 7

2.3 User Classes and Characteristics 9

2.4 Operating Environment 10

2.5 Design and Implementation Constraints 12

2.6 Assumptions and Dependencies 13

3. System Features 14

3.1 Order Placement 14

3.2 Delivery partner 15

3.2 Menu Manaagement 17

4. External Interface Requirements 18

4.1 User Interfaces 19

4.2 Hardware Interfaces 20

4.3 Software Interfaces 21

4.4 Communications Interfaces 22

5. Other Nonfunctional Requirements 22

5.1 Performance Requirements 23

5.2 Safety Requirements 23

5.3 Security Requirements 24

5.4 Software Quality Attributes 25

6. Other Requirements 25

Appendix A: Glossary 26

1. Introduction

1.1 Purpose

A multi-restaurant platform with comprehensive features serves as a centralized hub for culinary exploration and convenience. By amalgamating diverse eateries under one virtual roof, it offers a culinary journey with options galore. The platform's emphasis on best reviews ensures users make informed choices, fostering a community of discerning food enthusiasts. Seamless delivery services bring the gastronomic delights to the doorstep, transcending geographical constraints.

The extensive array of items and references caters to varied tastes, creating a gastronomic haven for everyone. Ratings not only guide choices but also cultivate a culture of excellence, encouraging restaurants to consistently deliver quality. This amalgamation of features transforms the dining experience into a dynamic, user-centric adventure, where exploration, reliability, and taste converge. In essence, it simplifies the complexities of dining choice, enriching lives through culinary diversity and elevating the enjoyment of food to new heights.

## 1.2 Document Conventions

The document conventions for a multi-restaurant platform with diverse features, including best reviews, efficient delivery services, a comprehensive list of items, and user ratings, are crucial for clarity and consistency. Adopting a standardized format for documentation ensures seamless communication and understanding across the development and user communities.

Utilizing a clear and uniform structure for feature descriptions, such as using headings, bullet points, and concise explanations, enhances readability. Clearly defined terminology and consistent naming conventions for features, reviews, delivery processes, and item references establish a shared language, reducing ambiguity.

Including visual elements like diagrams, flowcharts, and screenshots further elucidates complex functionalities, aiding both developers and end-users in comprehending the intricate aspects of the multi-restaurant platform. Additionally, maintaining an up-to-date document versioning system helps track changes, ensuring that all stakeholders are aligned with the latest developments.

In essence, a well-defined set of document conventions acts as a guiding framework, streamlining development, enhancing user understanding, and fostering effective collaboration in the evolution of the multi-restaurant platform.

## Intended Audience and Reading Suggestions

For a comprehensive documentation related to a multi-restaurant platform with features like best reviews, delivery services, and a wide variety of items, the intended audience and reading suggestions can be as follows:

**Restaurant Owners and Managers**

**Intended Audience**: Owners and managers of the restaurants integrated into the platform.

**Reading Suggestions**: Documentation on how to onboard their restaurant onto the platform, manage menu items, update pricing, and respond to customer reviews. Provide insights into analytics and reporting tools to help them optimize their offerings.

**Delivery Drivers:**

**Intended Audience**: Individuals involved in the delivery service.

**Reading Suggestions**: Training manuals for delivery drivers, including guidelines on order pickup, navigation, and customer interaction. Emphasize safety protocols and communication practices.

**Administrators and Support Teams:**

**Intended Audience:** Platform administrators and customer support teams.

**Reading Suggestions**: Administrative guides for managing the overall platform, handling support tickets, and resolving issues. Include troubleshooting guides for common user and restaurant-related issue

## Project Scope

The project scope for a multi-restaurant platform with features such as best reviews, delivery services, and a diverse range of items involves outlining the boundaries, objectives, and deliverables of the project. Here's an overview of the project scope:

**Platform Overview:**

Develop a multi-restaurant platform accessible via web and mobile applications.

Provide a user-friendly interface for customers, restaurant owners, delivery drivers, and administrators.

**User Features:**

Enable customers to register, log in, and create profiles.

Implement a robust search and filtering system for users to browse restaurants and menu items.

Allow users to leave reviews, ratings, and feedback for restaurants and items.

**Restaurant Management:**

Offer a streamlined onboarding process for restaurants to join the platform.

Provide tools for restaurant owners to manage their menus, update item details, and set pricing.

Implement analytics and reporting features for restaurant owners to monitor performance.

**Delivery Services:**

Facilitate a seamless order placement and payment process for users.

Integrate a real-time tracking system for customers to monitor the status of their orders.

Develop a robust delivery management system for dispatching and tracking delivery drivers.

**Item Catalog:**

Implement a comprehensive catalog system with categories, subcategories, and filters for a wide variety of items.

Include high-quality images and detailed descriptions for each menu item

* 1. **Reference**

•Deksne,et al[1] This system provides a comprehensive platform for managing orders, customizing menus, and overseeing various aspects of restaurant operations.

•Soon Nyean Cheong, et al[2][1] L. Deksne, A. Kempelis, T. Sniedzins, and A. Kozlovskis, “Automated System for Restaurant Services,” Inf. Technol. Manag. Sci., vol. 24, pp. 15–25, Dec. 2021, doi: 10.7250/itms-2021-0003.

•Soon Nyean Cheong, Wei Wing Chiew, and Wen Jiun Yap, “Design and development of Multi-touchable E-restaurant Management System,” in 2010 International Conference on Science and Social Research (CSSR 2010), IEEE, Dec. 2010, pp. 680–685. doi: 10.1109/CSSR.2010.5773867.

•Wei Wing Chiew, et al[2][1] L. Deksne, A. Kempelis, T. Sniedzins, and A. Kozlovskis, “Automated System for Restaurant Services,” Inf. Technol. Manag. Sci., vol. 24, pp. 15–25, Dec. 2021, doi: 10.7250/itms-2021-0003.

•[2] Soon Nyean Cheong, Wei Wing Chiew, and Wen Jiun Yap, “Design and development of Multi-touchable E-restaurant Management System,” in 2010 International Conference on Science and Social Research (CSSR 2010), IEEE, Dec. 2010, pp. 680–685. doi: 10.1109/CSSR.2010.5773867.

# 2.Overall Description

## Product Perspective

**Context and Origin:**

The content and origin of a multi-restaurant platform involve a combination of factors related to user needs, industry trends, technological innovations, and collaboration with various stakeholders.

**Content:**

**Restaurant Information**: Detailed information about the participating restaurants, including their names, cuisines, locations, and operating hours.

**Menus and Items**: Comprehensive listings of menu items, including detailed descriptions, prices, and images, showcasing the variety offered by each restaurant.

**Ordering**: Information on the ordering process, including a user-friendly interface, shopping cart details, and secure checkout procedures.

**Delivery Services**: Details on delivery options, estimated delivery times, and real-time tracking for users to monitor their orders.

**Origin:**

**Industry Best Practices**: Learning from successful multi-restaurant platforms and industry leaders helps in identifying best practices and innovative features that contribute to user satisfaction.

**Collaboration with Restaurants:** Content related to restaurant information, menus, and items originates from collaborating with restaurants, understanding their needs, and ensuring a smooth onboarding process onto the platform.

References for such platforms can be drawn from existing successful platforms like Uber Eats, Door Dash, Grubhub, or regional platforms like Swiggy and Zomato. Analyzing their content, user interfaces, and features can provide insights into industry standards and user expectations.

Staying informed about emerging trends in the food delivery and restaurant industry through industry reports, conferences, and publications contributes to the continuous improvement and innovation of the platform.

## 2.2 Product Features

The system comprises several key features organized into distinct modules to facilitate efficient operations for various users involved in the system. The major features include:

**User Authentication and Access Control:**

* Registration and login functionalities for Admin, Customers, Restaurants, and Delivery Partners.
* Role-based access control to ensure appropriate permissions and restrictions.

**Admin Management:**

* Special feature for admin to check the customers and restaurants on the restaurant.
* User management: Create, modify, and delete user accounts.
* Restaurant management: Adding, updating, or removing restaurant details.

**Customer Interface:**

* Account creation and profile management for customers.
* Browsing restaurants, viewing menus, and placing orders.
* Previous order history for the customers.

**Restaurant Management:**

* Menu creation, modification, and management.
* Order processing and assignment to delivery partners.
* Updating order status and delivery progress.

**Delivery Partner Operations:**

* Accepting and managing assigned delivery orders.
* updating status of the order placed by the customer.
* Cancellation of the order if the customer is unavailable or unreachable.

**Order Management System:**

Efficient order handling, including order placement, status updates, and cancellations.



## User Classes and Characteristics

**1. Admin**

**Characteristics:**

High privilege level with access to the entire system.

Responsible for system configuration, user management, and overseeing system operations.

Technical expertise in system administration and management.

Frequency of Use: Regularly, for system monitoring and management.

**2. Customer**

**Characteristics:**

Varied technical expertise, ranging from novice to experienced users.

Prioritize ease of use, intuitive interface, and efficient order processing.

Desire access to a wide range of restaurant options and user-friendly order placement.

Frequency of Use: Regularly, for browsing, ordering, and tracking orders.

**3. Restaurant**

**Characteristics:**

Moderate to advanced technical skills, primarily in handling the restaurant management interface.

Require efficient order processing, menu management, and order tracking capabilities.

Need to communicate order status and coordinate with delivery partners.

Frequency of Use: Frequently, for managing menu, processing orders, and coordinating deliveries.

**4. Delivery Partners**

**Characteristics:**

Basic technical skills for using the delivery app/interface.

Prioritize clear order details, navigation assistance, and timely order updates.

Focus on efficient order delivery and good communication with customers and restaurants.

Frequency of Use: Frequent, while actively delivering orders.

**5. System Guests (Unregistered Users)**

**Characteristics:**

Limited access to system features, primarily browsing restaurants and menus.

No profile creation or order placement capabilities until registration/login.

Intuitive and user-friendly interface preferred for easy navigation.

Frequency of Use: Occasional, for browsing or exploring available restaurants.

## Operating Environment

**Operating Systems and Software Dependencies:**

**Operating Systems:** The system is compatible with various operating systems including but not limited to:

Windows (Windows 10 and later versions)

macOS (macOS Catalina and later versions)

Linux distributions (Ubuntu, Fedora, CentOS)

**Software Dependencies:** The system relies on several software components and frameworks:

Java Runtime Environment (JRE): Required for executing the Java-based backend developed using Spring Boot.

Node.js: Required for running the frontend developed using React.js.

Database Management System (DBMS): Compatibility with MySQL 8.0 for data storage and retrieval.

**Network Requirements**

Internet Connectivity: The system operates over standard internet connections, requiring adequate bandwidth for seamless user interactions, real-time updates, and data transfers.

Security Protocols: Utilizes HTTPS (Hypertext Transfer Protocol Secure) to ensure encrypted communication between clients and the server, safeguarding sensitive data such as user credentials and payment information.

**Backend Framework:**

Spring Boot: Utilizing Java-based Spring Boot framework for backend server-side logic and API development.

Database: Connect database using MYSQL for data storage and retrieval.

Security: Integration with security protocols and libraries for user authentication and data encryption.

**Frontend Framework:**

React JS: Building the frontend interface for web and mobile applications.

Web Browsers: Compatibility with modern browsers such as Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.

## 2.5 Design and Implementation Constraints

**Design Constraints:**

**Responsive Design**: The user interface should be responsive, considering various devices such as desktops, tablets, and mobile phones.

**Cross-Browser Compatibility:** Ensure compatibility with major browsers (Chrome, Firefox, Safari, and Edge) to provide a consistent user experience.

**User Accessibility:** Design should adhere to accessibility standards, providing an inclusive experience for users with disabilities.

**Scalability**: The design should be scalable to accommodate potential growth in the number of restaurants, users, and transactions.

**Implementation Constraints:**

**Technology Stack Compatibility:** Ensure that React and Spring frameworks are compatible and can seamlessly communicate.

**Data Security**: Implement robust security measures for user data, including encryption for sensitive information and secure authentication mechanisms.

**Performance**: Implement optimizations for performance, considering factors like server response time, efficient database queries, and client-side rendering.

**Regulatory Compliance**: Adhere to relevant regulations and standards, such as data protection laws, especially if handling user information.

**Deployment Environment:** Define the deployment environment, considering factors like hosting services, server configurations, and version control.

## 2.6 Assumptions and Dependencies

**Assumptions:**

**Availability of Reliable Internet Connection:**

Assumption: It is assumed that all users (customers, restaurants, delivery partners) have access to a reliable internet connection for accessing and using the system.

**Compliance with Regulatory Standards:**

Assumption: The project assumes compliance with relevant data protection and privacy regulations concerning the handling of user data and transactions.

**Adequate Hardware and Software:**

Assumption: Users (restaurants and customers) have access to suitable hardware (computers, smartphones) and compatible software (browsers, operating systems) to interact effectively with the platform.

**Cooperation from Restaurants:**

Assumption: The success of the project depends on the willingness and active participation of restaurants to manage their menus, process orders, and collaborate with the platform.

**Dependencies:**

**Timely Development and Deployment:**

Dependency: The project's progress and success might depend on the timely completion of software development milestones and successful deployment of system updates or enhancements.

**Vendor Support and Service Availability:**

Dependency: The project might rely on external vendors or service providers for essential functions thus depending on their availability and support.

**User Adoption and Engagement:**

Dependency: The success of the project relies on user acceptance, adoption, and active engagement with the platform by both customers and restaurants.

**Regulatory Changes:**

Dependency: Any changes in data protection laws or industry regulations could impact the project, requiring adjustments to ensure compliance.

# System Features

**3.1 Order Placement**

The Order Placement feature is the fundamental functionality that empowers customers to interact with the system by selecting items from restaurant menus, creating orders, and finalizing transactions. This feature acts as the bridge between customers and restaurants, facilitating the seamless placing of orders for food items.

**3.1.1 Description and Priority**

**Description:**

**Customer Interaction:** The feature allows customers to explore menus from various restaurants listed on the platform.

**Order Customization:** Customers can add multiple items to their cart, adjust quantities, and remove items before confirming the order.

**Checkout Process:** After reviewing the cart, customers proceed to checkout, select payment options, and confirm the order.

**Confirmation of the order**: Upon successful order placement, customers can see the tracking of the order.

**Priority:**

The Order Placement feature holds a high priority within the system's functionalities due to its pivotal role in the user experience and revenue generation. It is a core feature influencing customer engagement and satisfaction, directly impacting the success of the platform. The smooth execution of this feature contributes significantly to user retention and overall system performance.

The priority is determined based on its crucial role in facilitating transactions and user engagement, ensuring it receives ample attention during development and testing phases to guarantee a robust and user-friendly functionality.

**3.1.2 Stimulus/Response Sequences**

**Stimulus 1:** Customer selects items from the menu and adds them to the cart.

System Response: Display updated cart with selected items and their quantities.

Additional Response: Display total order value, allowing the customer to review the cart before proceeding to checkout.

**Stimulus 2:** Customer proceeds to checkout and confirms the order.

System Response: Confirmation message acknowledging the order placement and estimated delivery time.

Additional Response: Initiate payment gateway for transaction processing.

**3.1.3 Functional Requirements**

**REQ-1:** Display restaurant menus with categorized items and prices.

Ensure a user-friendly interface displaying menus with clear categories, item descriptions, prices, and available customizations.

**REQ-2:** Allow customers to add items to the cart and adjust quantities.

Enable customers to easily add/remove items, modify quantities, and manage their cart contents.

**REQ-3:** Provide a checkout process allowing customers to review the order and confirm.

Create a step-by-step checkout process including order review, address selection, and payment confirmation.

**REQ-4:** Calculate estimated delivery time based on restaurant preparation and delivery distance.

**3.2 Delivery Partner Assignment**

The Delivery Partner Assignment feature is the backbone of the order fulfillment process within the multi-restaurant system. Upon order placement by a customer, this functionality orchestrates the intelligent allocation of suitable delivery partners to ensure prompt, efficient, and reliable delivery of orders from restaurants to customers' locations.

**3.2.1 Description and Priority**

**Description:**

**Real-time Matching:** The system swiftly identifies available delivery partners near the restaurant and assesses their workload and proximity to efficiently assign the most appropriate partner.

**Transparency and Customer Satisfaction:** Offers transparency to customers and restaurants by displaying the assigned delivery partner and estimated delivery times, instilling confidence and reliability in the delivery process.

**Priority:**

The Delivery Partner Assignment feature holds a High priority within the system architecture. Its significance lies in directly impacting the timely and satisfactory delivery of orders, thereby influencing customer satisfaction and retention.

This priority is a reflection of its pivotal role in optimizing delivery logistics, ensuring smooth operations, and upholding the system's credibility.

**3.2.2 Stimulus/Response Sequences**

**Stimulus 1**: Order Placed by Customer

System Response: Identifies available delivery partners based on proximity to the restaurant and current workload.

**Stimulus 2:** System Initiates Delivery Partner Assignment

System Response: Utilizes an algorithm to assign the most suitable available delivery partner based on criteria such as distance, workload, and previous performance.

**3.2.3 Functional Requirements**

**REQ-1**: Location-based Partner Identification

Identify and retrieve the list of available delivery partners in proximity to the restaurant.

**REQ-2:** Workload Assessment

Calculate the workload of available delivery partners to determine their availability for new orders.

**REQ-3**: Update on the website when the order is received

Automatically notify the customer about the order and update the order status in the website.

**3.3 Menu Management**

**3.3.1 Description and priority**

**Description:**

**Menu Updates**: Equips restaurant staff with a feature to effortlessly add new items, edit existing ones, and remove outdated or unavailable items from their menu listings.

**Rich Item Customization:** Enables detailed customization of menu items, including titles, descriptions, prices, images, and availability status, ensuring accuracy and attractiveness to customers.

**Real-time Visibility**: Reflects instant changes made by restaurant staff across the platform, ensuring customers have access to the most current menu offerings.

**Enhanced Customer Engagement:** Allows restaurants to categorize items for easier navigation, enhancing the browsing experience for customers and influencing their ordering choices.

**Priority:**

The Menu Management feature is designated as a High priority element within the system architecture. Its significance lies in its direct influence on customer engagement, ordering decisions, and overall user satisfaction.

**3.3.2 Stimulus/Response Sequences**

**Stimulus 1:** Restaurant staff adds a new menu item.

System Response: Validates the input, updates the menu database, and displays the new item in the restaurant's menu list.

**Stimulus 2**: Restaurant staff edits an existing menu item.

System Response: Allows modifications to the item's details (price, description, availability), updating the database accordingly.

**Stimulus 3:** Restaurant staff removes a menu item.

System Response: Deletes the item from the menu database and updates the menu list for customers.

**3.3.3 Functional Requirements**

**REQ-1:** Menu Item Addition

Provide an interface for restaurant staff to add new menu items, including details such as name, description, price, and availability status.

**REQ-2:** Menu Item Editing

Allow restaurant staff to modify existing menu items, including price adjustments, description changes, and availability updates.

**REQ-3:** Menu Item Deletion

Enable restaurant staff to remove items from the menu, ensuring that the deletion process is intuitive and prompt.

**REQ-4:** Image Upload for Menu Items

Allow the addition of images for menu items to enhance visual appeal for customers browsing the menu.

**REQ-5:** Menu Category Management

Provide options for restaurant staff to organize menu items into categories (e.g., appetizers, mains, desserts) for easy navigation.

# External Interface Requirements

## User Interfaces

**Customer Interface**

* Customer Home page: Provides access to account information, order history, and a search/browse feature for restaurants and menu items.
* Menu Viewing: Displays restaurant menus with item details, prices, images, and add-to-cart functionality.
* Order Placement: Allows customers to add items to their cart, proceed to checkout, and confirm orders.
* Order History: Interface to view the past orders the customer has placed .

**Admin Interface**

* Restaurant Management: Interface for adding, editing, or removing restaurants from the platform, along with their details and menus.
* User Management: Enables the management of customer accounts, delivery partner profiles, and staff access rights.
* Order Management: Facilitates viewing and managing orders, resolving disputes, and tracking overall order performance.

**Restaurant Interface**

* Menu Management: Allows restaurant staff to add, update, or remove menu items, set prices, manage availability, and upload images.
* Order Processing: Interface for accepting, preparing, and updating the status of incoming orders.
* Performance Analytics: Provides insights into popular items and customer feedback for improvement.

**Delivery Partner Interface**

* Order Assignment: Enables the acceptance or rejection of incoming delivery requests and provides navigation tools for efficient order delivery.
* Order Status Updates: Allows updating order statuses, indicating pickup, en-route, and order delivery completion.

**Common Elements Across Interfaces**

* Consistent Design Language: Adherence to a unified design theme, ensuring consistency and familiarity across different interfaces.
* Navigation Bar: A standard navigation menu or bar providing easy access to key sections within each interface.
* Search and Filter Options: Tools for searching, sorting, and filtering content to enhance usability.

## Hardware Interfaces

**Supported Device Types**

* Desktop Computers/Laptops: The software product should be compatible with standard desktops and laptops running popular operating systems (e.g., Windows, macOS, Linux) for administrative tasks, customer ordering, and restaurant management.
* Smartphones and Tablets: Support for iOS and Android devices for customer-facing applications, allowing users to place orders, track deliveries, and access restaurant menus.

**Nature of Interactions**

* Data Interactions: Bidirectional data flow between the software and hardware components for transmitting and receiving information related to orders, menus, user accounts, and real-time updates.
* Control Interactions: Software control over hardware functionalities such as utilizing cameras for image uploads.

**Communication Protocols**

* Web-based Protocols: HTTP/HTTPS protocols for web-based interactions between the software and web browsers on various devices.

**Additional Considerations**

* Screen Resolutions: Support for various screen resolutions and aspect ratios to ensure a consistent and responsive user interface across different device types.
* Input Methods: Adaptation to different input methods (touchscreens, keyboards, mice) for ease of use and accessibility across devices.

## Software Interfaces

**User Authentication and Authorization Interface:**

* Purpose: Manages user login, authentication, and authorization processes across different user roles.
* Components: Utilizes authentication protocols securely authenticate users and manage their access rights based on their roles.

**Database Interface:**

* Purpose: Stores and manages various data entities such as user profiles, restaurant details, menu items, orders, delivery information, and more.
* Components: Interacts with a relational database management system using SQL queries to retrieve, insert, update, and delete data.

**Menu Management Interface:**

* Functionality: Facilitates the management and display of menus for restaurants within the system.

**Components:**

* Allows restaurant owners to create, update, and manage their menus, including adding/removing items, setting prices, and specifying availability.
* Provides an interface for users (customers) to browse through restaurant menus, view item details, and make selections.

**Ratings and Reviews Interface:**

* Functionality: Enables users (customers) to rate and review restaurants and their food items.

**Components:**

* Allows users to submit ratings and reviews for restaurants based on their dining experiences, food quality, delivery times, etc.
* Displays aggregated ratings and reviews for each restaurant to assist other users in making informed decisions.

**Admin Dashboard Interface:**

* Functionality: Provides an administrative interface for system administrators to manage the entire platform.

**Components:**

* Enables system administrators to monitor and manage user accounts, access permissions, and system configurations.
* Provides analytics and reporting tools for insights into system usage, user behaviors, and performance metrics.
  1. **Communication Interfaces**

**Order Status Notifications**

Functionality: Sends basic order status notifications to users.

Components:

* Notifies users about order confirmations and updates on order preparation and delivery status.
* Provides simple alerts to users regarding any changes in the order status or estimated delivery times.

**System-generated Alerts**

Functionality: Provides basic alerts within the system interface.

Components:

* Displays alerts within the platform for order confirmations, updates, or important announcements to users.
* Alerts restaurant owners and delivery partners about new orders or changes directly within the system interface.

**Basic Announcements**

Functionality: Sends general announcements or updates.

Components:

* Provides basic announcements within the platform interface for general information relevant to users and stakeholders.
* Offers updates about system maintenance or policy changes directly within the system.

**5.Other Non-Functional Requirements**

**5.1 Performance Requirements**

**Response Time:**

Requirement: The system should respond to user interactions (e.g., browsing menus, placing orders) within an average response time of 2 seconds.

**System Availability:**

Requirement: The system should maintain at least 99.9% uptime, allowing access to functionalities and services 24/7, excluding scheduled maintenance windows.

**Scalability:**

Requirement: The system should accommodate a concurrent user load of 1000 active users without a degradation of performance during peak hours.

**Database Performance:**

Requirement: Database queries for retrieving menu information or processing orders should execute within 300 milliseconds on average.

**Peak Load Handling:**

Requirement: The system should handle a 50% increase in user traffic during promotional events or peak usage times without a significant increase in response time.

**5.2 Safety Requirements**

**Data Protection for Menus and Orders:**

Requirement: Ensure encryption and secure storage of restaurant menus, order histories, and proprietary information to prevent unauthorized access or data breaches.

**Secure Account Management:**

Requirement: Provide customers with secure account management features, including strong password requirements and options for multi-factor authentication.

**Secure Access Controls:**

Requirement: Implement robust access controls for restaurant owners/managers, allowing access only to relevant data and functionalities within the system.

**Secure Communication:**

Requirement: Enable secure communication channels between restaurants and the platform, ensuring confidentiality and integrity of communications regarding orders and updates.

**5.3 Security Requirements**

**Data Encryption:**

Requirement: All sensitive data, including user credentials, payment information, and personal details, must be encrypted using industry-standard encryption during transmission and storage.

**Secure User Authentication:**

Requirement: Implement secure authentication mechanisms (e.g., multi-factor authentication) for users accessing the system to prevent unauthorized access and protect user accounts**.**

**Access Control:**

Requirement: Enforce strict access controls to restrict system functionalities based on user roles (customers, restaurant owners, delivery partners, admins), ensuring that each user can only access authorized features.

**Data Backup and Recovery:**

Requirement: Implement regular data backup mechanisms and ensure disaster recovery procedures are in place to prevent data loss and enable quick system recovery in case of failures or emergencies.

**7. Privacy Protection:**

Requirement: Ensure compliance with privacy regulations by providing users with control over their personal data, obtaining consent for data processing, and maintaining transparent data handling practices.

**5.4 Software Quality Attributes**

Software quality attributes for a multi-restaurant platform are pivotal for delivering a seamless and reliable experience to users and stakeholders.

**Reliability**: Ensuring consistent and dependable performance is crucial. Users rely on the platform for accurate order processing, timely deliveries, and real-time updates. Reliability builds trust and customer satisfaction.

**Scalability**: The platform should be designed to handle varying loads seamlessly, accommodating fluctuations in user traffic, order volumes, and restaurant partnerships. Scalability ensures optimal performance during peak times and future growth.

**Usability**: An intuitive and user-friendly interface is essential for both customers and restaurant owners. Clear navigation, easy menu customization, and efficient order management contribute to a positive user experience.

**Security**: Robust data protection measures, secure payment gateways, and compliance with privacy regulations are imperative. Users must trust that their personal and financial information is handled with the utmost security.

**Performance Efficiency**: Efficient data processing, quick response times, and minimal downtime contribute to overall performance efficiency. Load balancing mechanisms and optimized algorithms enhance the platform's operational speed.

**Flexibility**: The ability to adapt to changing business requirements, technological advancements, and diverse restaurant models is crucial. A flexible platform can integrate new features seamlessly and evolve with the dynamic food industry landscape.

By prioritizing these software quality attributes, a multi-restaurant platform can establish a solid foundation for delivering a reliable, user-friendly, and adaptable service that meets the needs of both customers and restaurant partners.

**6.Other Requirements**

**Appendix A: Glossary**

**Terms**

Multi-Restaurant Platform: The overarching system that aggregates and connects multiple restaurants, providing users with a centralized platform for exploring and ordering from various establishments.

**Vendor/Restaurant Partner**: Individual restaurants or food providers that join the platform to showcase their offerings.

**Menu Aggregation:** The process of compiling menus from different restaurants into a cohesive and easily navigable interface for users.

**User Profile:** An individual's account on the platform, containing personal information, order history, and preferences.

**Order Management System (OMS**): The backend system responsible for handling and processing customer orders, managing communication between users and restaurants.

**Delivery Service**: The option for users to have their food orders delivered to a specified location, often facilitated by third-party delivery services or an in-house delivery team.

**Rating and Reviews**: User-generated feedback and evaluations of restaurants and their products, providing insights and guidance for other users.

**Cuisine Categories**: Different food types or styles, helping users to filter and find specific types of restaurants or dishes.

**Dashboard**: The interface, often separate for both users and restaurants, where they can manage their accounts, view orders, and access relevant information.

**Acronyms and Abbreviations**

**MRS** - Multi-Restaurant System

**RMS** - Restaurant Management System

**CRS** - Customer-restaurant System

**DPS** - Delivery Partner System

**UI** - User Interface

**DBMS** - Database Management System

**QA** - Quality Assurance

**POS -** Point of Sale

**JSON** - JavaScript Object Notation

**HTTPS** - Hypertext Transfer Protocol Secure