## Analysis and Specify Software Quality Requirements

### Security and Privacy Protection

Must Needs:-

- Authentication and authorization, this include secure authentication mechanisms for all users. Furthermore using multi factor authentication for not just customers but for staff and administrators.
- Access Control is vital for implementing where users are authorised to go. Customers can submit feedback, staff can moderate and respond to feedback and administrators have oversight for the system so that they may view logs.
- Data Privacy Compliance is critical in following regulations such as GDPR and CCPA for public visible reviews allowing for anonymizing personal data to protect customer identities.
- Data Encryption! There are 2 main types with encryption of data, cryptographic protocols (SSL/TLS) as well as symmetric encryption(AES-256).

### Performance

Must Needs:-

- Fast Image Processing: Use image compression techniques to reduce file sizes for uploaded photos, ensuring they load quickly without compromising quality which will be essential due to customers taking pictures of their meals as well as drinks.
- Efficient Filtering and Sorting: Allow users to quickly filter and sort feedback by criteria like date, rating, or helpfulness. Use database indexing to optimise query performance for frequently used filters.
- Caching of Frequently Accessed Feedback: Implement caching for commonly viewed feedback entries (e.g., the latest reviews or most helpful comments) to reduce database load and improve response time for end-users.
- Optimised Feedback Display and Submission: Minimise loading times when customers view, submit, or modify feedback. This includes optimising database queries and using pagination for lists of reviews to avoid overloading the system.

### Reliability

Must Needs:-

- Error handling in feedback submission and moderation must be included to implement robust error-handling mechanisms to further ensure that customers and staff receive clear error messages if there any issues during the submission, moderation or viewing.
- Consistent feedback availability will ensure that feedback data is always accessible, even during peak/busy times to outages. By implementing this, it will minimise downtime.
- Automated backup paired with quick recovery is essential to minimise the loss of data as well as recovery of data. Establishing a good recovery plan will enable the restoration of lost data quickly and maintain reliability

**Scalability**

Must Needs:-

- Database Sharding for large feedback volumes: database sharding to distribute feedback data across multiple servers if the volume of feedback grows significantly. This will prevent bottlenecks and ensure that queries remain efficient even with a high volume of data.
- Cloud storage scalability for uploaded images: Store uploaded feedback images in a scalable cloud storage solution, allowing for easy storage expansion as the number of images grows. Use a content delivery network (CDN) to serve images efficiently.
- Autoscaling During High Demand Periods: Enable autoscaling for the feedback subsystem during peak usage periods to handle sudden increases in user activity, such as after promotional events or during holidays when feedback volume may spike.

**Search for Restaurants**

Security and Privacy Protection

- Secure Location Services: User location data must be transmitted via encrypted protocols (e.g., HTTPS) and anonymized if stored or used for analytics. This keeps locations and data search more secure.
- Input Sanitization: They must prevent malicious queries through strict input validation and protection against SQL injection.

Performance

- Optimized Search Queries: Implement geospatial indexing and keyword optimization to return search results in under 2 seconds. Search should feel very quick showing the results quicker.
- Autocomplete Caching: Frequently searched names and filters should be cached for fast suggestion rendering. Popular searches should be quick to further save time.

Reliability

- Fault-Tolerant Filtering: In case of service outages (e.g., map or cuisine filter), return partial results with appropriate error notices.
- Redundant Servers: Ensure 24/7 availability by hosting search services on high-availability infrastructure.

Scalability

- Scalable Search Index: Use distributed search services (like Elasticsearch) to scale based on region or city growth.
- Auto-Scaling Backend: Automatically scale the number of search microservices based on peak meal-time traffic.

**Booking a Table**

Security and Privacy Protection

- Session Security: Use token-based session validation and timeout policies to protect booking transactions.

- Booking Access Control: Bookings should be visible only to the customer and restaurant staff with proper role-based access. The only people to see the booking should only be the person who made the booking as well as the required staff. Sessions need to be monitored as well as protected.

Performance
- Real-Time Table Availability: Use in-memory data stores (e.g., Redis) for fast updates on table availability. Bookings should be quick with live updates on table availability.
- Streamlined Booking Flow: Booking should complete within 5 steps or less, optimized for both speed and usability.

Reliability
- Concurrency Control: Ensure double-booking is prevented using database locks or atomic transactions. (operations that are treated as a single, indivisible unit of work)
- Retry and Confirmation: Failed booking attempts should automatically retry with clear error feedback or fallback suggestions.

Scalability
- Load-Balanced Services: Deploy booking microservices with load balancers to distribute traffic.
- Live Availability Sync: Table status must reflect real-time changes across multiple users via WebSockets or polling.
- The system must also handle more booking during busy hours such as weekends/holidays.

**Pre-Order of Food**

Security and Privacy Protection
- Data Confidentiality: Pre-order details, including allergy or dietary preferences, must be securely stored and transmitted. They should at all times be private.
- Authenticated Orders Only: Only authenticated users can place pre-orders, with secure validation tokens.

Performance
- Menu Optimization: Cache daily menus and compress images to improve menu load times.]
- Quick-Select Pre-Orders: Enable favorites and recent orders for faster repeat purchases.

Reliability
- Persistent Storage: Ensure pre-orders are saved even in case of app or network failures, using local backup or queues.
- Acknowledgment Protocols: Pre-orders must be confirmed with server-side receipts, and re-confirmed upon check-in.
- Once saved they must be done properly in the event of a crash the database should remain correct

Scalability

- Dynamic Menu Management: Use modular services to handle regional menus, allergen data, and special offers.
- High-Volume Order Handling: Enable burst traffic support with auto-scaling containers and message queuing systems (e.g., Kafka or RabbitMQ).

**Customer Feedback**

Security and Privacy Protection

- Authentication & Authorization: Secure feedback systems using multi-factor authentication for all users and role-based access control.
- Data Privacy Compliance: Feedback must comply with GDPR/CCPA, anonymizing user data in public reviews. Once a review has been made the users information must remain private.
- Encryption: Use SSL/TLS for transmission and AES-256 for storage of user-submitted feedback and metadata.

Performance

- Fast Image Processing: Compress feedback images before upload to reduce load times.
- Optimized Querying: Use indexed searches and caching to support sorting/filtering by rating, date, or usefulness.
- Responsive Display: Paginate feedback results to reduce client-side rendering and backend load.

Reliability

- Error Handling: Show user-friendly messages for failed uploads or invalid entries, with retry options.
- High Availability: Feedback should remain accessible at all times, with backup systems in place.
- Automated Backup & Recovery: Ensure frequent backups and fast recovery to preserve submitted data.

Scalability

- Distributed Storage: Store feedback across sharded databases and scalable cloud solutions.
- Autoscaling on High Volume: Enable auto scaling during peak feedback periods (e.g., post-events or holidays).
- CDN Integration: Use CDNs to serve review images efficiently worldwide.
- As more users leave reviews as well as upload images, the system should also grow with demand.