**Detailed Explanation of Each Requirement**

**Functional Requirements (FRs):**

**FR1: Customers shall be able to make table reservations online and in person.**

* + This feature allows customers to book tables through a web or mobile app or directly at the restaurant. It improves convenience and reduces wait times.

**FR2: The system shall track walk-in customers and assign tables dynamically.**

* + This ensures efficient seating arrangements by matching walk-in customers to available tables based on party size and table availability.

**FR3: Waiters shall input customer orders through a digital interface.**

* + Waitstaff will use a device (e.g., tablet or POS system) to take orders, which reduces errors and speeds up communication with the kitchen.

**FR4: Orders shall be relayed to the kitchen in real time.**

* + Orders are sent instantly to the kitchen once entered, reducing delays and ensuring prompt food preparation.

**FR5: Kitchen staff shall update order statuses in real time.**

* + The kitchen can notify waitstaff when an order is ready, streamlining operations and minimizing confusion.

**FR6: The system shall track inventory levels and notify managers when stock is low.**

* + Inventory tracking ensures that ingredients are available, and automated alerts help prevent stockouts.

**FR7: The system shall allow the manager to update the menu dynamically.**

* + Managers can easily add, remove, or modify menu items (e.g., for seasonal specials or sold-out dishes).

**FR8: The system shall manage employee scheduling and shifts.**

* + This feature enables managers to assign shifts, handle absences, and optimize staff schedules.

**FR9: The system shall generate financial reports (daily, weekly, and monthly).**

* + Automated financial reporting provides insights into revenue, expenses, and profitability.

1. **FR10: Users shall have different roles and permissions (waiters, chefs, managers).**
   * Role-based access control ensures that users can only access features relevant to their responsibilities.
2. **FR11: The system shall support customer loyalty programs.**
   * Tracks customer visits, purchases, and rewards points, encouraging repeat business.

**FR12: The system shall allow customers to view live waiting times.**

* + Provides real-time updates on table availability, reducing frustration and improving customer experience.

**FR13: The system shall enable customers to leave feedback and rate their dining experience.**

* + Customers can share their experiences, helping the restaurant improve service and identify issues.

**FR14: The system shall generate personalized promotions and discounts.**

* + Uses customer data to offer tailored deals, boosting engagement and sales.

**FR15: The system shall include a feature for handling online food delivery orders.**

* + Supports delivery services by integrating order placement, tracking, and customer notifications.

**FR16: The system shall allow staff to manage multiple languages.**

* + Improves accessibility for diverse customer bases, enabling better communication and service.

**FR17: The system shall provide a feature to split bills between multiple customers.**

* + Simplifies payments for groups, ensuring flexibility and convenience.

**FR18: The system shall integrate with payment gateways for digital transactions.**

* + Enables smooth payments via credit/debit cards, e-wallets, or other digital payment methods.

 **FR19: The system shall allow staff to manage table layouts and seating arrangements.**

* Enables customization of the restaurant's floor plan for better management of space and special events.

 **FR20: The system shall allow integration with third-party food delivery platforms.**

* Supports linking with popular services like Uber Eats, DoorDash, etc., for external order handling.

 **FR21: The system shall allow managers to set and monitor sales targets for employees.**

* Provides tools for setting sales goals and tracking progress, improving employee performance.

 **FR22: The system shall allow customers to pre-order food before arriving at the restaurant.**

* Speeds up service for busy customers and ensures kitchen readiness for specific requests.

 **FR23: The system shall track customer preferences and dietary restrictions.**

* Records details such as allergies or food preferences for a personalized experience.

 **FR24: The system shall allow staff to process refunds and handle disputes.**

* Simplifies the resolution of payment issues or dissatisfaction with orders.

 **FR25: The system shall enable notification alerts for pending tasks (e.g., unpaid bills, delayed orders).**

* Ensures that no task is overlooked, improving service efficiency.

 **FR26: The system shall track energy and utility usage in the restaurant.**

* Helps monitor operating costs and identify opportunities for cost-saving.

 **FR27: The system shall provide performance analytics for staff and menu items.**

* Offers insights into employee efficiency and the popularity of dishes.

 **FR28: The system shall support seasonal or promotional menu updates with start and end dates.**

* Automates temporary menu changes for holidays or events.

 **FR29: The system shall allow customers to customize their orders.**

* Provides flexibility to adjust ingredients or portions for individual preferences.

 **FR30: The system shall integrate with a queue management system.**

* Optimizes waiting lines for customers, whether seated or for takeout orders.

Non-Functional Requirements (NFRs):

 **NFR1: The system shall have a response time of less than 2 seconds for most operations.**

* Ensures quick system performance, enhancing user satisfaction.

 **NFR2: The system shall ensure secure access with authentication and authorization.**

* Protects sensitive data and restricts access to authorized users only.

 **NFR3: The system shall be accessible on both mobile and desktop devices.**

* Ensures compatibility across platforms for staff and customers.

 **NFR4: The system shall provide automatic data backups at regular intervals.**

* Prevents data loss by creating regular backups.

 **NFR5: The system shall scale seamlessly to handle high traffic.**

* Ensures smooth operation during peak hours or special events.

 **NFR6: The system shall comply with relevant data protection and privacy regulations.**

* Adheres to laws like GDPR to safeguard customer data.

 **NFR7: The system shall ensure compatibility with popular POS hardware.**

* Works efficiently with existing restaurant hardware, avoiding additional costs.

 **NFR8: The system shall log all user actions for auditing purposes.**

* Tracks user activity for security and compliance purposes.

 **NFR9: The system shall provide an intuitive user interface.**

* Simplifies system use, reducing training requirements and user errors.

 **NFR10: The system shall support multilingual user interfaces for staff and customers.**

* Facilitates usage by a diverse workforce and customer base.

 **NFR11: The system shall provide downtime notifications and alternative procedures.**

* Ensures minimal disruption during system maintenance or outages.

 **NFR12: The system shall support cross-platform data synchronization.**

* Keeps data consistent across mobile apps, desktop systems, and POS devices.

 **NFR13: The system shall maintain an uptime of 99.9%.**

* Guarantees high availability to support continuous operations.

 **NFR14: The system shall encrypt all sensitive customer data.**

* Ensures that customer information is protected against breaches.

 **NFR15: The system shall allow exporting data in multiple formats (e.g., PDF, CSV).**

* Provides flexibility for sharing or analyzing reports externally.

 **NFR16: The system shall allow customization of system themes and branding.**

* Adapts to the restaurant's branding needs for a professional look.

 **NFR17: The system shall ensure real-time synchronization for multi-location restaurants.**

* Keeps data updated across multiple branches, ensuring consistency.

 **NFR18: The system shall provide automated reminders for maintenance (e.g., cleaning kitchen equipment).**

* Helps the restaurant stay compliant with health and safety standards.

 **NFR19: The system shall allow integration with loyalty program APIs.**

* Supports seamless operation with third-party loyalty tools.

 **NFR20: The system shall allow reporting of environmental sustainability metrics.**

* Tracks metrics like waste reduction and energy efficiency for eco-friendly initiatives.

**Detailed breakdown** of high-priority functional and non-functional requirements

**High-Priority Functional Requirements**

**FR1: Customers shall be able to make table reservations online and in person.**

* **Sub-requirements:**
  1. Design a user-friendly online interface for reservations.
  2. Implement a calendar-based scheduling system to manage reservation slots.
  3. Integrate in-person reservation tracking with the same database.
* **Implementation Steps:**
  1. Create a database table for reservations (fields: customer name, contact, date, time, table number, special requests).
  2. Develop APIs for adding, updating, and canceling reservations.
  3. Design a web interface and integrate it with mobile platforms for customer use.
* **Testing Considerations:**
  1. Stress test with simultaneous reservation requests.
  2. Validate time conflicts and table availability.

**FR2: The system shall track walk-in customers and assign tables dynamically.**

* **Sub-requirements:**
  1. Create a dashboard for table status (available, occupied, reserved).
  2. Develop logic to assign tables based on party size and availability.
* **Implementation Steps:**
  1. Use algorithms to match customer group size with table capacity.
  2. Include override functionality for manual table assignments by staff.
* **Testing Considerations:**
  1. Test dynamic table assignment during peak hours.
  2. Handle scenarios of walk-ins when all tables are occupied.

**FR4: Orders shall be relayed to the kitchen in real time.**

* **Sub-requirements:**
  1. Provide an order interface for waiters and customers (e.g., kiosks or apps).
  2. Implement notifications to the kitchen as soon as an order is placed.
* **Implementation Steps:**
  1. Use WebSocket or similar technologies for real-time communication.
  2. Develop a kitchen interface to display orders with time stamps and priorities.
* **Testing Considerations:**
  1. Simulate multiple concurrent orders and ensure they are relayed without delay.
  2. Test kitchen notification accuracy under different network conditions.

**FR6: The system shall track inventory levels and notify managers when stock is low.**

* **Sub-requirements:**
  1. Set thresholds for low inventory alerts.
  2. Provide reports for inventory usage trends.
* **Implementation Steps:**
  1. Create a database table for inventory (fields: item name, quantity, reorder level).
  2. Implement backend logic to send alerts when stock falls below thresholds.
  3. Integrate with vendor management for restocking.
* **Testing Considerations:**
  1. Test low-stock notifications with various threshold levels.
  2. Ensure no false-positive or missed alerts.

**FR9: The system shall generate financial reports (daily, weekly, and monthly).**

* **Sub-requirements:**
  1. Include revenue, expenses, and profits in the reports.
  2. Enable export to formats like PDF or Excel.
* **Implementation Steps:**
  1. Design database queries to aggregate financial data.
  2. Use a reporting library to generate formatted outputs.
  3. Schedule automatic report generation with email notifications.
* **Testing Considerations:**
  1. Verify report accuracy with test financial data.
  2. Ensure proper formatting and compatibility with Excel and PDF.

**FR10: Users shall have different roles and permissions (waiters, chefs, managers).**

* **Sub-requirements:**
  1. Define role-based access control (RBAC) for different user types.
  2. Provide an admin panel to manage roles and permissions.
* **Implementation Steps:**
  1. Create a user table with role fields.
  2. Implement middleware to check permissions for specific actions.
  3. Develop role management functionality in the admin dashboard.
* **Testing Considerations:**
  1. Test role-based restrictions for all system features.
  2. Attempt unauthorized actions to ensure proper blocking.

**High-Priority Non-Functional Requirements**

**NFR1: The system shall have a response time of less than 2 seconds for most operations.**

* **Sub-requirements:**
  1. Optimize database queries and API calls.
  2. Use caching for frequently accessed data.
* **Implementation Steps:**
  1. Use performance monitoring tools to identify bottlenecks.
  2. Implement database indexing and efficient query structures.
* **Testing Considerations:**
  1. Conduct load testing with tools like JMeter.
  2. Monitor response times under peak traffic conditions.

**NFR2: The system shall ensure secure access with authentication and authorization.**

* **Sub-requirements:**
  1. Use industry-standard encryption for passwords and sensitive data.
  2. Implement multi-factor authentication (MFA) for admin accounts.
* **Implementation Steps:**
  1. Use libraries like JWT (JSON Web Token) for session handling.
  2. Encrypt sensitive data using AES-256 or similar algorithms.
  3. Integrate MFA APIs.
* **Testing Considerations:**
  1. Conduct penetration testing to identify vulnerabilities.
  2. Validate that permissions are correctly enforced for each role.

**NFR5: The system shall scale seamlessly to handle high traffic during peak hours.**

* **Sub-requirements:**
  1. Use cloud infrastructure for scalable backend services.
  2. Implement load balancing for traffic distribution.
* **Implementation Steps:**
  1. Deploy services on cloud platforms (e.g., AWS, Azure).
  2. Use horizontal scaling for database and application servers.
* **Testing Considerations:**
  1. Simulate high traffic scenarios and measure system performance.
  2. Ensure no downtime or performance degradation during scaling.

**NFR13: The system shall maintain an uptime of 99.9%.**

* **Sub-requirements:**
  1. Use redundant servers to minimize downtime.
  2. Schedule maintenance during off-peak hours.
* **Implementation Steps:**
  1. Set up automated monitoring for server health.
  2. Use a disaster recovery plan for quick restoration of services.
* **Testing Considerations:**
  1. Conduct failover tests to verify redundancy.
  2. Simulate hardware and network failures.