Faculty of Cyber Physical System
Dept. of IoT and Robotics Engineering (IRE)

Project Proposal

Project title: Cloud Kitchen Management System

Course Tile: Software Engineering Lab
Course code: ICT 4354

Submitted by:
Md. Tanjib Riasat (ID: 2001008)
Chayon Kumar Das (ID: 2001015)
Suvro Kumar Das (ID: 2001021)

Submitted to
Shifat Ara Rafiq
Lecturer,
Dept. of SE, BDU.

Session: 2020-2021
Level- 3 **Term-** 1
Batch: 3rd

Submission Date: September 24, 2024

Requirement Specification of Cloud Kitchen Management System

Software Requirements Specification (SRS)

The Software Requirements Specification (SRS) document outlines the requirements for the **Cloud Kitchen Management System**, covering specific subsystems or features. It is intended for various audiences, including developers, project managers, testers, marketing staff, end users, and documentation writers. The document is structured for easy navigation, starting with an overview and moving through sections like functional requirements, non-functional requirements, and user interface details.

Document Organization

This SRS is structured to allow each reader type to focus on the most relevant sections. The recommended sequence for reading is as follows:

- **Overview:** Begin with the introduction and project scope to gain a high-level understanding.
- **Functional Requirements:** Developers and testers should proceed to this section for detailed functional specifications.
- **Non-functional Requirements:** Read for performance, reliability, and usability requirements.
- **User Interface Requirements:** For user experience-related concerns.
- **Appendices and Glossary:** For reference to any terms or concepts not fully explained in the core sections.

System Requirements

System requirements outline the hardware, software, and infrastructure necessary for the product to function correctly. These requirements ensure that the system operates as intended and meets both user and technical needs.

System Requirements for Cloud Kitchen Management System

Hardware Requirements

- **Processor:** Minimum Dual-core, 2.0 GHz or higher.
- **RAM:** Minimum 4 GB.
- **Storage:** 256 GB SSD for fast data access and secure storage.
- **Network Interface:** Gigabit Ethernet for reliable connectivity.
- **Server Environment:** Cloud-based infrastructure with auto-scaling capabilities.

Software Requirements

- **Web Server:** Apache/Nginx for hosting the application.
- **Database:** MySQL/PostgreSQL for data management.
- **Backend Framework:** Node.js/Django/Flask for server-side logic.
- **Frontend Framework:** React/Vue.js/Angular for user interface.
- **Payment Gateway Integration:** Secure payment processing solution.

Network Requirements

- **Secure Connection:** TLS/SSL for secure data transmission.
- **Firewalls:** Implement on both server and client sides to prevent unauthorized access.
- **Load Balancer:** To distribute incoming traffic evenly across servers.

Functional Requirements

Functional requirements describe the specific behaviors and functionalities that the system must exhibit to meet the user's needs. These include inputs, processes, outputs, and interactions within the system.

Functional Requirements for Cloud Kitchen Management System

1. **User Registration:**
    - Allow users to create accounts on the platform.
    - Collect necessary information such as email address, password, and contact details.
    - Provide options for social media login (if applicable).
2. **Restaurant Registration:**
    - Allow restaurant owners to create virtual restaurant profiles.
    - Collect necessary information such as restaurant name, location, cuisine type, and contact details.
3. **Menu Management:**
    - Enable restaurants to add, edit, and delete food items from their menus.
    - Provide options for specifying item details, pricing, and descriptions.
4. **Food Item Browsing:**
    - Allow users to search for food items by category, cuisine, or keyword.
    - Provide a filter option to narrow down search results based on location (division or district).
5. **Restaurant Profiles:**
    - Display detailed information about individual restaurants, including menu, ratings, and reviews.
6. **Ordering System:**

    - Facilitate online food ordering directly from the platform.
    - Integrate with payment gateways for secure transactions.
    - Provide options for delivery or pickup.
7. **Customer Reviews and Ratings:**
    - Allow customers to rate and review restaurants and their food items.
    - Display average ratings and reviews on restaurant profiles.
8. **Promotions and Discounts:**
    - Enable restaurants to offer promotions, discounts, or loyalty programs.
    - Provide a platform for displaying and managing these offers.
9. **Notifications:**
    - Send notifications to customers about order status, promotions, and new menu items.
    - Notify restaurants about new orders or inquiries.
10. **Analytics and Reporting:**
    - Provide analytics on restaurant performance, popular food items, and customer behavior.
    - Generate reports for restaurant owners to help them make data-driven decisions.

Non-Functional Requirements

Non-functional requirements define the quality attributes and performance standards of the system, ensuring it operates effectively and efficiently.

Non-Functional Requirements for Cloud Kitchen Management System

1. **Performance:**
    - **Response Time:**
        - Use efficient algorithms for database queries, search operations, and order processing to minimize latency.
        - Implement caching mechanisms (e.g., Memcached, Redis) for faster data retrieval.
    - **Hardware:**
        - Ensure the underlying hardware can handle the expected load, potentially scaling resources as needed.
2. **Scalability:**
    - **Architecture:**
        - Design the system to be horizontally scalable, allowing for the addition of more servers to handle increased load.
    - **Database:**
        - Use a scalable database solution (e.g., MySQL, PostgreSQL, MongoDB) that can handle large datasets and high transaction rates.
3. **Maintainability:**

- **Microservices:**
  - Consider a microservices architecture to break down the system into smaller, independent components that can be scaled individually.
- **Load Balancing:**
  - Implement load balancing techniques to distribute traffic evenly across multiple servers.

4. **Reliability:**
   - **Redundancy:**
     - Implement redundancy in critical components (e.g., servers, databases) to ensure continued operation during failures.
   - **Monitoring:**
     - Use monitoring tools to track system performance and identify potential issues proactively.
   - **Backup and Disaster Recovery:**
     - Regularly backup data and have a disaster recovery plan in place.
5. **Secure Authentication:**
   - **Password Hashing:**
     - Use strong password hashing algorithms (e.g., bcrypt, Argon2).
   - **Two-Factor Authentication (2FA):**
     - Implement 2FA to enhance security.
6. **Secure Authorization:**
   - **Role-Based Access Control (RBAC):**
     - Enforce RBAC to restrict access to system resources based on user roles and permissions.
7. **Reliable Communication:**
   - **Data Encryption:**
     - Encrypt sensitive data both at rest and in transit.
   - **Secure Communication:**
     - Use HTTPS to protect against eavesdropping.
   - **Regular Security Audits:**
     - Conduct audits to identify and address vulnerabilities.
   - **Web Application Firewall (WAF):**
     - Deploy a WAF to protect against common web-based attacks.
8. **Usability:**
   - **Intuitive Interface:**
     - Design a user-friendly interface with clear labeling and consistent design.
   - **User Testing:**
     - Conduct testing to gather feedback for improvements.
   - **Error Handling:**
     - Provide clear error messages to guide users.
9. **Accessibility:**
   - **WCAG Compliance:**
     - Adhere to Web Content Accessibility Guidelines to ensure usability for individuals with disabilities.

- **Alternative Text and Keyboard Navigation:**
  - Provide alternative text for images and ensure full keyboard navigation.
- **Color Contrast and Screen Reader Compatibility:**
  - Ensure sufficient color contrast and test compatibility with screen readers.