**Software Requirements Specification (SRS) for Pantry2Plate**

4. Non-Functional Requirements

Non-functional requirements are quality attributes that describe the overall behavior and performance characteristics of the system. These requirements are crucial to ensure that the Pantry2Plate system is robust, secure, and user-friendly.

4.1 Performance

4.1.1 Response Time

The system should respond to user interactions (e.g., loading the pantry inventory, searching for recipes, creating meal plans) within 2 seconds.

4.1.2 Concurrent Users

The system should support a minimum of 1000 concurrent users without significant degradation in performance.

4.1.3 Scalability

The system should be designed to scale horizontally to accommodate an increasing number of users and data.

4.1.4 Database Performance

Database queries and transactions should be optimized to ensure rapid data retrieval and updates.

4.2 Security

4.2.1 Data Encryption

User data, including passwords and personal information, should be securely stored and transmitted using strong encryption protocols (e.g., TLS).

4.2.2 Authentication

User authentication should follow industry best practices, including the use of salted and hashed passwords.

4.2.3 Authorization

Users should only have access to data and features they are authorized to use.

4.2.4 Session Management

User sessions should be securely managed to prevent unauthorized access.

4.2.5 Audit Trail

The system should maintain an audit trail of user actions for security and accountability purposes.

4.3 Usability

4.3.1 User Interface

The user interface should be intuitive, user-friendly, and responsive, ensuring a positive user experience.

4.3.2 Accessibility

The system should be designed to be accessible to users with disabilities, following relevant accessibility guidelines (e.g., WCAG).

4.4 Reliability

4.4.1 Uptime

The system should have a minimum uptime of 99.9% to ensure it is available to users 24/7.

4.4.2 Backup and Recovery

Regular backups of the database and user data should be performed, and a reliable data recovery plan should be in place.

4.4.3 Error Handling

The system should gracefully handle errors and provide clear error messages to users.

4.5 Compliance

4.5.1 Legal and Regulatory Compliance

The system should comply with relevant data protection and privacy laws (e.g., GDPR, HIPAA) depending on the user's location and data handled.

4.5.2 Internationalization

The system should be designed to support multiple languages and regions.

4.6 Documentation

4.6.1 User Documentation

Provide user-friendly documentation, including user guides and FAQs, to assist users in utilizing the system effectively.

4.6.2 Technical Documentation

Create comprehensive technical documentation for developers and maintainers, including architecture diagrams and code documentation.

4.7 Performance Testing

4.7.1 Load Testing

Perform load testing to verify the system's ability to handle a high volume of users and data.

4.7.2 Security Testing

Conduct security testing, including vulnerability assessments and penetration testing, to identify and mitigate security risks.

4.7.3 Usability Testing

Engage users in usability testing to gather feedback on the system's user interface and overall user experience.