**EXPERIMENT 2**

**NON-FUNCTIONAL REQUIREMENTS**

Non-functional requirements, also known as quality attributes or constraints, define the characteristics and constraints of the system beyond its functionality. These requirements describe how the system should perform, rather than what it should do. Non-functional requirements are often related to performance, reliability, security, usability, and other aspects that contribute to the overall system quality. Examples include response time, system availability, data encryption, user interface design, and regulatory compliance.

* **Performance**

The platform must provide quick response times, especially during real-time updates.

Load testing should be conducted to ensure optimal performance under varying user loads.

* **Security**

Robust security measures must be in place to protect user data and privacy.

Encryption protocols should be implemented for secure data transmission.

* **Scalability**

The system should be designed to handle a growing number of users and recipes.

Scalability testing should be performed to assess the platform's ability to expand.

* **Usability**

The user interface should be intuitive, accommodating users with varying levels of technical expertise.

User experience testing should be conducted to ensure ease of use.

* **Reliability**

The platform must be reliable, minimizing downtime and service interruptions.

Implementing backup and recovery mechanisms is essential for data integrity.

* **Regulatory Compliance**

The system must comply with relevant data protection and privacy regulations.

Transparency in terms of service and privacy policies is crucial for user trust.