**Functional Requirements:**

Functional requirements define **what a system should do**. They describe the specific behaviors, tasks, or functionalities of the system to meet the user's needs. These are often framed as use cases or user stories and directly address the business goals of the system.

**Examples:**

1. **User Authentication:** The system must allow users to log in using a username and password.
2. **Payment Processing:** The system must support credit card and PayPal transactions.
3. **Data Retrieval:** Users should be able to search for and retrieve records using specific filters.
4. **Notifications:** The system should send email notifications for account updates.
5. **Role Management:** Admin users should be able to add, delete, or modify user roles.

---

**Non-Functional Requirements:**

Non-functional requirements define **how a system should behave** or **the quality attributes of the system**. These focus on performance, reliability, usability, and other aspects that make the system effective and user-friendly.

**Examples:**

1. **Performance:** The system must handle up to 10,000 concurrent users without degradation.
2. **Security:** Data must be encrypted in transit and at rest.
3. **Usability:** The interface should be accessible to visually impaired users (e.g., compliant with WCAG standards).
4. **Scalability:** The system should scale to accommodate a 50% increase in traffic each year.
5. **Availability:** The system should be available 99.99% of the time, excluding scheduled maintenance.
6. **Maintainability:** Developers should be able to update the system with minimal

downtime.

7. **Portability:** The application must run on both Windows and macOS.

---

**Key Differences:**

| Aspect | Functional Requirements | Non-Functional Requirements |
|---|---|---|
| **Focus** | What the system should do | How the system should perform |
| **Type** | Features and functionalities | Quality attributes |
| **Measurement** | Pass/fail based on functionality | Measured against benchmarks |
| **Examples** | Login, payment, search functionality | Performance, reliability, usability |

Both are essential for building a successful system! Functional requirements ensure the system does the right tasks, while non-functional requirements ensure it does them efficiently and effectively.