

Example of Functional and Non-Functional Requirements

Scenario: An Online Banking Application

1. Functional Requirements

These are the features or functions the system must perform. They define what the software should do.

- **User Login:**
 - The system must allow users to log in using their username and password.
 - **Funds Transfer:**
 - The system should enable users to transfer funds between their accounts.
 - **Balance Inquiry:**
 - The user must be able to view their account balance.
 - **Bill Payments:**
 - The system must allow users to pay utility bills online.
 - **Transaction History:**
 - Users should be able to view their transaction history for the past 6 months.
-

2. Non-Functional Requirements

These define how the system performs a function and are often called quality attributes.

- **Performance:**
 - The system should process user requests within 2 seconds.
- **Security:**
 - All sensitive user data must be encrypted, and multi-factor authentication should be used for user logins.
- **Reliability:**
 - The system must be available 99.9% of the time, excluding scheduled maintenance.
- **Scalability:**
 - The application must be able to handle 10,000 concurrent users.
- **Usability:**

- The user interface should be intuitive and user-friendly, with clear navigation and instructions.

- **Compliance:**

- The system must comply with banking regulations and data protection laws.
-

Summary

- **Functional Requirements:** What the system should do (specific functions/features).
- **Non-Functional Requirements:** How the system should behave (quality and performance standards).