Non-Functional Requirements:

1. **General Types/Categories of Non-Functional Requirements:**

The non-functional requirements that will be followed can include categories such as:

1. Performance: Concerned with the system's responsiveness, throughput, and resource utilization under various conditions.
2. Reliability: Focuses on the system's ability to maintain its functionality and data integrity over time, including fault tolerance and recovery mechanisms.
3. Security: Encompasses measures to protect the system from unauthorized access, data breaches, and malicious attacks.
4. Usability: Addresses the system's ease of use, accessibility, and user experience to ensure efficient interaction with the platform.
5. Scalability: Refers to the system's ability to handle increasing workload and user base without significant performance degradation.
6. Maintainability: Deals with the ease of maintaining, updating, and extending the system over its lifecycle, including modularity and documentation.
7. Compatibility: Ensures that the system can operate seamlessly with other hardware, software, and platforms as per defined standards.
8. Compliance: Concerned with meeting regulatory requirements, industry standards, and organizational policies relevant to the system's operation.

**b) Non-Functional Requirements Specification:**

| **Category** | **Requirement Description** |
| --- | --- |
| Performance | The system shall maintain an average response time of less than 2 seconds for course page loading. |
| Reliability | The system shall perform regular data backups and ensure data integrity with a backup frequency of at least once per day. |
| Security | The system shall implement HTTP for secure data transmission between the server and client. |
| Usability | The system shall provide a user-friendly interface with intuitive navigation and clear instructional prompts. |
| Scalability | The system shall support a minimum of 1000 concurrent users without degradation in performance. |
| Maintainability | The system shall adhere to coding standards and provide comprehensive documentation for ease of maintenance and future development. |
| Compatibility | The system shall be compatible with the latest versions of major web browsers including Chrome, Firefox, Safari, and Edge. |
| Compliance | The system shall comply with GDPR regulations regarding the collection and handling of user data. |

**c) Fit Criteria for Every Non-Functional Requirement (Testable Non-Functional Requirements):**

* Performance: Measure average response time using load testing tools and compare against the specified requirement.
* Reliability: Verify data backup processes and perform restoration tests to ensure data integrity.
* Security: The system shall implement HTTP for secure data transmission between the server and client.
* Usability: Conduct user testing sessions and gather feedback on interface intuitiveness and navigation.
* Scalability: Perform stress testing to determine the system's capacity to handle concurrent users.
* Maintainability: Review codebase for adherence to coding standards and evaluate completeness of documentation.
* Compatibility: Test system functionality across multiple web browsers and platforms to ensure compatibility.

d) **Impact on the System's Overall Architecture:** Each non-functional requirement can significantly influence the design and architecture of the system:

* Performance requirements may necessitate optimizations such as caching mechanisms, and database indexing to ensure responsiveness.
* Reliability requirements may require redundant server configurations, failover mechanisms, and automated backup systems to maintain data integrity.
* Security requirements may influence the implementation of authentication protocols, access controls, encryption algorithms, and intrusion detection systems within the system's architecture.
* Usability requirements may impact the layout, navigation flow, and information presentation within the user interface, influencing the choice of design patterns and interaction paradigms.
* Scalability requirements may drive the adoption of distributed architectures, microservices, and cloud-based infrastructure to accommodate growth in user base and workload.
* Maintainability requirements may guide the use of modular, loosely coupled components and the adoption of version control and continuous integration practices to facilitate system maintenance and updates.
* Compatibility requirements may influence technology stack choices, API design, and integration strategies to ensure interoperability with external systems and platforms.
* Compliance requirements may necessitate the implementation of data protection measures, audit trails, and regulatory reporting mechanisms within the system's architecture to ensure adherence to legal and regulatory standards.