**Non-Functional Requirements**

**1. Scalability:**

- Horizontal Scalability: The system should be able to handle a growing number of users, tweets, and interactions by adding more nodes to the blockchain network.

- Vertical Scalability: The system should be able to handle increased load by upgrading hardware or software components.

**2. Performance:**

- Throughput: The blockchain should support a high transaction throughput to handle tweets and interactions efficiently.

- Latency: The system should provide low latency for posting tweets and retrieving them, ensuring a responsive user experience.

**3. Security:**

- Data Encryption: Use encryption techniques to secure user data, private keys, and sensitive information.

- Consensus Algorithm: Choose a secure consensus algorithm (e.g., Proof of Work or Proof of Stake) to maintain the integrity of the blockchain.

- Identity Management: Implement robust identity and access management to prevent unauthorized access to user accounts.

- Smart Contract Security: Ensure smart contracts are secure and audit them for vulnerabilities before deployment.

**4. Reliability:**

- Fault Tolerance: The system should be able to withstand node failures or network partitions without compromising data integrity.

- Data Redundancy: Implement data redundancy and backup mechanisms to prevent data loss.

**5. Availability:**

- High Availability: Ensure the system is available 24/7 with minimal downtime for maintenance.

- Load Balancing: Use load balancing to distribute incoming traffic evenly across nodes.

**6. Consistency:**

- Blockchain Consistency: Guarantee the consistency of the blockchain ledger across all nodes in the network.

- Tweet Consistency: Ensure tweets and interactions are consistent and not lost or duplicated.

**7. User Experience:**

- User Interface Responsiveness: Ensure that the user interface is responsive and easy to use.

- Transaction Confirmation: Provide timely feedback on transaction status to users.

**8. Interoperability:**

- Compatibility: Ensure that the blockchain system can interact with other platforms and services, facilitating integration with external apps and tools.

**9. Backup and Recovery:**

- Data Backup: Regularly backup blockchain data to prevent data loss in case of catastrophic events.

- Disaster Recovery: Develop a disaster recovery plan to quickly restore the system in case of failures.