**Block chain-Based Digital Identity Verification**

Zainab Eman | Imama Kainat

22F3738 | 22F2661

2024

Contents

[1. Performance Requirements Response Time: 1](#_Toc183503311)

[Throughput: 1](#_Toc183503312)

[Scalability: 1](#_Toc183503313)

[Availability: 1](#_Toc183503314)

[Benchmarking Tools: 1](#_Toc183503315)

[2. Security Requirements Data Security: 1](#_Toc183503316)

[API Security: 1](#_Toc183503317)

[Access Control: 1](#_Toc183503318)

[Penetration Testing: 2](#_Toc183503319)

[Compliance: 3](#_Toc183503320)

[3. Non-Functional Requirements Scalability: 3](#_Toc183503321)

[Reliability: 3](#_Toc183503322)

[Usability: 3](#_Toc183503323)

[Maintainability: 3](#_Toc183503324)

[Auditability: 3](#_Toc183503325)

[4. Detailed Test Plan Document Functional Testing: 3](#_Toc183503326)

[Performance Testing: 3](#_Toc183503327)

[Security Testing: 3](#_Toc183503328)

[Integration Testing: 3](#_Toc183503329)

[Edge Case Testing: 3](#_Toc183503330)

[Automation Strategy: 3](#_Toc183503331)

[5. Software Test Plans Purpose: 4](#_Toc183503332)

[Scope: 4](#_Toc183503333)

[Environment: 4](#_Toc183503334)

[Tools: 4](#_Toc183503335)

[Defect Reporting: 4](#_Toc183503336)

[Test Metrics: 4](#_Toc183503337)

Requirements for the Blockchain-Based Digital Identity Verification Project

# Performance Requirements Response Time:

 All primary API endpoints involved in identity creation, verification,

and management should consistently respond within **1 second** under standard operating conditions.

# Throughput:

 The platform is designed to handle a minimum of **500 concurrent API**

**requests** while maintaining performance integrity, ensuring latency does not increase by more than **200 milliseconds**.

# Scalability:

 The system architecture must accommodate horizontal scaling to support anticipated user growth, handling up to **10,000 active users per hour**

without degradation in service quality.

# Availability:

 High availability is critical, with the system maintaining **99.9% uptime**. This will be achieved through robust redundancy mechanisms and failover solutions to ensure continuous operation.

# Benchmarking Tools:

 Employ **Locust** and **JMeter** for stress testing the system under various load conditions to ensure it meets performance benchmarks.

 Regular monitoring of blockchain transaction finality times and smart contract gas usage to optimize performance and cost-efficiency.

# Security Requirements Data Security:

 All sensitive user data, including biometric identifiers, must be hashed and encrypted using industry-standard algorithms such as **SHA-256 for**

**hashing** and **AES-256 for encryption** before storage.

 Real-time processing of biometric data ensures that no sensitive

information is ever permanently stored, mitigating the risk of data breaches.

# API Security:

 Implement **OAuth2** framework to ensure secure and robust handling of authentication and authorization processes.

 Enforce **HTTPS** to secure API communications and use input validation to guard against common vulnerabilities like injection attacks and invalid data submissions.

 Implement nonce mechanisms to protect against replay attacks,

particularly in interactions involving blockchain transactions.

# Access Control:

 Implement a detailed **Role-Based Access Control (RBAC)** system, allowing users to manage cryptographic key-based permissions effectively.

 Provide mechanisms for users to dynamically audit and revoke third-party access to their identity data as needed.

# Penetration Testing:

 Conduct comprehensive security assessments using tools like **OWASP ZAP**

and **Burp Suite**, along with custom scripts designed to test the

robustness of the system against potential security threats.

# Compliance:

 Adhere strictly to **GDPR** guidelines for data privacy and **ISO/IEC 27001**

standards for security management, ensuring the system is compliant with legal and industry requirements.

# Non-Functional Requirements Scalability:

 The platform is architected to gracefully handle increases in load and

transaction volume, ensuring a seamless user experience as the user base grows.

# Reliability:

 Utilize the inherent immutability of blockchain technology to provide a secure and tamper-proof repository for identity records, ensuring data integrity even in adverse conditions.

# Usability:

 Design user interfaces for web and mobile applications to be intuitive and easy to navigate, allowing users to manage their digital identities without technical expertise.

# Maintainability:

 Code should be modular and well-documented, adhering to best practices

to facilitate future updates and maintenance without disrupting existing functionality.

# Auditability:

 Implement comprehensive logging and auditing capabilities to record all transactions and interactions with user data, providing transparent and immutable audit trails.

# Detailed Test Plan Document Functional Testing:

 Thoroughly validate functionality across all API endpoints to ensure accurate performance in scenarios such as identity creation,

verification, and revocation.

# Performance Testing:

 Conduct rigorous testing to assess system performance under various

simulated conditions to validate response times, throughput, and system scalability.

# Security Testing:

 Perform in-depth security testing to identify potential vulnerabilities and ensure the system's resilience against various types of cyber

threats.

# Integration Testing:

 Ensure seamless integration between the front-end applications, the backend APIs, and the blockchain infrastructure.

# Edge Case Testing:

 Test the system's behavior under unusual conditions or with atypical data inputs to ensure robust error handling and graceful degradation.

# Automation Strategy:

 Develop and implement a comprehensive automation strategy using tools like **Postman** for API testing and **RestAssured** for integration testing, integrating these into a continuous integration/continuous deployment (CI/CD) pipeline for ongoing validation.

# Software Test Plans Purpose:

 Validate that the blockchain-based identity verification system

functions as expected, meets security standards, and performs efficiently across various scenarios.

# Scope:

 Covering functional, performance, security, and edge case testing across all components of the system.

# Environment:

 Testing environments will include a **development setting** with simulated blockchain nodes, a **staging environment** mirroring live data, and a

**production environment** with full-scale deployment.

# Tools:

 Utilize a range of tools tailored to different testing needs, including

**Postman** for API functionality, **JMeter** and **Locust** for performance, and

**OWASP ZAP** and **Burp Suite** for security.

# Defect Reporting:

 Implement a systematic approach to defect tracking using tools like **JIRA**

or **Trello**, enabling efficient issue resolution and accountability.

# Test Metrics:

 Maintain meticulous records of test coverage, execution times, and defect rates to continually assess and improve the testing process.

These enhanced and detailed requirements are designed to guide the development and testing of the Blockchain-Based Digital Identity Verification project, ensuring a

robust, secure, and user-friendly system. Let me know if there's any particular area you'd like to expand further or adjust according to specific project needs.