System Group Test Documentation V0

# 1. Test Plan

## 1.1 Test Objectives

Ensure the RESTful API interfaces built by the System group are correct, secure, stable, and scalable, while validating their data flow and service interaction with the UI, Data, and Analysis groups.

## 1.2 Involved Modules (Detailed Breakdown)

|  |  |
| --- | --- |
| Module Name | Responsibility Description |
| API Service Module | Provides all REST APIs (including version control & docs) |
| Data Transfer Module | Receives data from UI → Forwards to Data and Analysis groups |
| Security Module | User authentication (JWT) and permission control (Spring Sec.) |
| Logging/Monitoring | System log collection (ELK) & performance metrics (Prometheus) |
| Deployment Module | CI/CD scripts with Jenkins/GitLab + Docker image packaging |

## 1.3 Test Resources

- Technical Stack: Java 17 + Spring Boot 3.x + Redis + Kafka + MySQL

- Test Tools: Postman, pytest, JUnit, Spring Test, Prometheus

- Data Sources: Simulated JSON payloads, tokens, header parameters

# 2. Test Strategy

## 2.1 Test Types Overview

Functional, security, exception, integration, performance, logging, and deployment tests using various tools like Postman, pytest, Spring Test, and Jenkins.

## 2.2 Risk Identification

|  |  |  |
| --- | --- | --- |
| Risk Type | Description | Mitigation Measures |
| API Exception | Unauthorized users can access sensitive APIs | Enforce JWT validation + unify error codes |
| Middleware Error | Kafka push failure or timeout | Add retry logic / circuit breaker mechanism |
| Data Inconsistency | Delay between data forwarding and analysis | Use async processing + message trace ID |
| CI/CD Crash | Image build or deployment failures | Add build logs and validation scripts |

# 3. Test Case Design

|  |  |  |  |
| --- | --- | --- | --- |
| TC ID | Description | Request Method | Expected Response |
| TC-API-001 | Successful user login returns JWT token | POST /api/login | 200 OK, response contains "token" field |
| TC-SEC-002 | Unauthorized access to protected endpoint | GET /api/user/info | 401 Unauthorized, error message returned |
| TC-DATA-003 | Data submission from UI forwarded to Data group | POST /api/data | 202 Accepted, Kafka logs input record |
| TC-ANL-004 | Analysis result successfully forwarded to frontend | GET /api/result | 200 OK, JSON result returned |
| TC-EXCP-005 | Missing data fields trigger appropriate error | POST /api/data | 400 Bad Request, missing field prompt |
| TC-CACHE-006 | User token is correctly stored in Redis | Redis inspection | Key exists with valid user JSON info |
| TC-DEPLOY-007 | CI/CD auto deployment triggered correctly | Git push → Jenkins | Build success, image uploaded successfully |

# 4. Test Report Outline

|  |  |
| --- | --- |
| Project Component | Content Description |
| Test Version | System Test V0.1 |
| Testers | System Group QA |
| Test Tools | Postman, pytest, Jenkins, Redis-cli |
| Test Execution | Total: 15 cases → Passed: 12, Failed: 3, Blocked: 0 |
| Issues Logged | JWT permission issues; old containers not auto-removed during deployment |
| Optimization Tips | Enhance log levels; improve exception handling and branch coverage |