# Test Scenario Template

| | |
|---|---|
| **Portfolio / Collection** | **Quality Assurance Document Portfolio** |
| **Document Classification** | **Enterprise-Grade QA workflow** |
| **Author** | **Sashika Samaragunaratne** |
| **Role** | **Senior Test Analyst / QA Lead** |
| Purpose Statement | This document is part of a curated Quality Assurance documentation portfolio demonstrating enterprise-level QA practices across requirement validation, risk-based testing, execution governance, and release decision support.<br>The artefacts reflect real-world QA leadership experience in:<br>• Multi-team delivery environments<br>• Requirement-light Agile programs<br>• Regulated and compliance-sensitive domains<br>• AI-assisted and data-driven systems |
| Intended Audience | Senior Test Analyst/ QA Leads, Product Owners, Engineering Managers, Architects |
| Domain/ Operating Content | Enterprise, multi-team systems operating in regulated environments, including healthcare compliance and AI-assisted decision platforms. |
| Document Owner | QA Lead/ Test Lead |
| Version Information | Version: 1.0<br>Status: Portfolio Sample<br>Last Updated: 27-JAN-2026 |
| Confidentiality Disclaimer | All examples, workflows, and data contained in this document are fictional and provided solely for professional portfolio demonstration purposes. |
| Usage Rights | This document may be reused or adapted for educational and professional purposes with appropriate attribution. |

# Table of Contents

# 1. Introduction

This document defines the **standard structure and expectations** for creating test scenarios within the QA framework.

Test scenarios created using this template:

- Are derived only from **validated requirements**
- Focus on **behaviour and outcomes**, not implementation
- Support manual execution, automation assessment, and audit evidence
- Are understandable by QA, Product, and Engineering teams

# 2. Scenario Design Principles

- Each scenario validates **one primary behaviour**
- Scenarios are written from a **business and risk perspective**
- UI mechanics are secondary to outcome validation
- Assumptions are explicitly referenced
- Scenarios must be executable, traceable, and reviewable

**Key Principle:**
A test scenario is a **communication artefact**, not a script.

# 3. Scenario Metadata

Scenario ID
Jira Reference (Story / Epic ID)
Scenario Type (Functional / Integration / API / Data / ML)
Priority (High / Medium / Low)
Risk Level (High / Medium / Low)
QA Owner

# 4. Test Scenario Definition

## Scenario Title

A clear, concise, behaviour-focused title.

## Business Context

A short description explaining **why this behaviour matters** from a business or compliance perspective.

Example:
Ensure compliance documents are validated correctly to prevent ineligible records from being accepted.

## Preconditions

Conditions that must be true before execution.

Example:

- User has valid access permissions
- System is operational
- Required reference data exists

## Scenario Steps (High-Level)

Steps describe **what happens**, not UI clicks.

Example:

1. Submit a compliance document with mandatory fields populated
2. Trigger document validation

### Expected Outcome

Observable results that determine pass/fail.

Example:

- Document status is updated to “Validated”
- No validation errors are displayed
- Audit log entry is created

# 5. Data and Validation Rules

### Data Requirements

Input Data: Valid / Invalid / Boundary
Reference Data: Configuration or lookup values
Data Source: Synthetic or Masked

### Validation Rules

Explicit rules being validated.

Example:

- Mandatory fields must not be null
- Document type must be supported
- Validation applies only to active records

# 6. Negative and Edge Case Coverage

Related scenarios or variations to be considered.

Examples:

- Missing mandatory fields
- Unsupported document type
- Expired or inactive record

# 7. Compliance and Regulatory Considerations

If applicable, document regulatory relevance.

Example:
This scenario validates behaviour required for regulatory audit and must retain processing history.

# 8. Automation Assessment

Automation Candidate: Yes / No
Automation Priority: High / Medium / Low
Automation Notes: Stability, data dependency, constraints

**QA Rule:**
Do not automate unstable or ambiguous behaviour.

# 9. Traceability

Requirement: Jira Story ID
Acceptance Criteria: AC reference
Assumption: If applicable
Test Plan: Individual Test Plan reference

# 10. Execution and Review

Execution Status: Not Run / Pass / Fail / Blocked
Defect Reference: Jira ID
Execution Notes: Observations

Reviewer Notes:
Space for review comments, clarifications, or follow-up actions.

# 11. Usage Guidelines

- One scenario per primary behaviour
- Avoid combining unrelated validations
- Keep language neutral and observable
- Update scenarios when requirements change
- Ensure traceability before execution