Test Plan for SOAP Request Project:

Calculator

**Created by: Poorani Santhosh**

### **1. Objective**

This document outlines the test plan for the SOAP Request Project - Calculator application. The objective is to ensure the web service accurately performs arithmetic operations (Addition, Subtraction, Multiplication, and Division) and functions correctly under various conditions. The target audience is developers, testers, and consumers of SOAP-based web services.

### **2. Scope**

**Features to be tested**:

SOAP API endpoint: <http://www.dneonline.com/calculator.asmx>

Functionality to perform:

* Addition of two integers
* Subtraction of two integers
* Multiplication of two integers
* Division of two integers.

**Types of testing**:

* Manual Testing
* Automated API Testing (e.g., Postman, REST Assured)
* Performance Testing (e.g., with JMeter)
* Accessibility and Compliance Testing (API schema validation).

**Environments**:

* Windows 10, macOS
* Browsers for testing API UI (Postman, SOAP UI)
* Devices: Desktop/Laptop
* Network: Wi-Fi and VPN connections

**Evaluation criteria**:

* Accuracy of arithmetic result
* Response time and uptime of API
* Number of defects found
* Time to resolve defects
* Feedback from stakeholders

**Team roles and responsibilities**:

* QA Lead: Planning, test case reviews, defect triage, reporting.
* QA Engineer: Test execution, defect logging, retesting.

### **3. Inclusions**

* **Introduction**: SOAP-based calculator API validation
* **Test Objectives**:
  + Ensure arithmetic operation accuracy
  + Validate boundary conditions and edge cases
  + Verify compliance with the WSDL schema.

### **4. Exclusions**

* Internal implementation of the API (black-box testing only)
* Security/penetration testing
* Testing of other operations.

### **5. Test Environments**

* **Operating Systems**: Windows 10, macOS Ventura
* **Browsers**: Postman, SOAP UI, Chrome (Dev Tools)
* **Devices**: Laptops (i5/i7, 8GB RAM minimum)
* **Network Connectivity**: Wi-Fi, VPN
* **Hardware/Software Requirements**: 8 GB RAM, Java 11, SOAP UI/Postman
* **Security Protocols**: No authentication required.
* **Access Permissions**: Open access, no login credentials required.

### **6. Defect Reporting Procedure**

* **Criteria**: Incorrect word result, unexpected error codes, invalid SOAP responses.
* **Reporting Steps**:
  + Defect Title, Steps to Reproduce, Actual vs Expected Result
  + Attach SOAP request/response logs/screenshots.
* **Triage and Prioritization**:
  + P1: Critical functionality failure (eg : incorrect division result)
  + P2: Incorrect numerical output formatting
  + P3: Minor layout or typo.
* **Tools**: JIRA
* **Roles**:
  + QA Engineer: Logs bugs
  + QA Lead: Reviews and assigns.
* **Communication**: Slack + Daily Status Emails
* **Metrics**:
  + No. of test cases executed
  + Defect Density
  + Test Case Pass/Fail Rate.

### **7. Test Strategy**

**Step 1: Test Scenario & Case Creation**

* **Techniques**:
  + Equivalence Partitioning (e.g., <1000, large integers, zero values)
  + Boundary Value Analysis (0, 1, negative integers)
  + Use Case Testing (common input sets).
* **Additional**:
  + Exploratory Testing
  + Error Guessing (e.g., division by zero).

**Step 2: Testing Procedure**

* Smoke Testing
* Functional Testing
* Retesting and Regression Testing
* Cross-platform validation via different SOAP clients
* End-to-End user flow validation.

**Step 3: Best Practices**

* Shift-Left Testing: Early validation of WSDL and sample response
* Context-Driven Testing
* Continuous Feedback Loop.

### **8. Test Schedule**

| **Task** | **Owner** | **Start Date** | **End Date** |
| --- | --- | --- | --- |
| Test Planning | QA Lead | 01-May-2025 | 02-May-2025 |
| Test Scenario Design | QA Team | 03-May-2025 | 04-May-2025 |
| Test Case Creation | QA Team | 05-May-2025 | 06-May-2025 |
| Test Execution | QA Engineer | 07-May-2025 | 12-May-2025 |
| Defect Logging & Retesting | QA Team | 08-May-2025 | 13-May-2025 |
| Test Report & Sign-Off | QA Lead | 14-May-2025 | 15-May-2025 |

### **9. Test Deliverables**

* Test Plan
* Test Scenarios
* Test Cases
* Daily Test Execution Reports
* Defect Report
* Final Test Summary Report.

### **10. Entry and Exit Criteria**

**Requirement Analysis**

* **Entry**: API Documentation (WSDL + sample requests)
* **Exit**: Test scope finalized.

**Test Execution**

* **Entry**: Reviewed test cases and test data
* **Exit**: All test cases executed, bugs logged.

**Test Closure**

* **Entry**: Defect resolution complete
* **Exit**: Final summary report shared.

### **11. Tools**

* **JIRA**: Defect Tracking
* **SOAP UI / Postman**: Manual Test Execution
* **Apache JMeter**: Performance Testing
* **Excel/Word**: Documentation
* **Snipping Tool**: Screenshots.

### **12. Risks and Mitigations**

| **Risk** | **Mitigation** |
| --- | --- |
| QA resource unavailable | Backup resource arrangement |
| Network/API downtime | Use offline WSDL or mock responses |
| Incomplete API Documentation | Direct communication with dev/stakeholder |

### **13. Approvals**

**Documents for Client Approval**:

* Test Plan
* Test Scenarios & Test Cases
* Test Execution Reports
* Final Summary Report.