**Test Plan for SOAP Request Project:**

**Number to Words**

Created by: [Kumari Rupam]

**1. Objective**

This document outlines the test plan for the Number to Words SOAP Web Service application (URL: <https://www.dataaccess.com/webservicesserver/NumberConversion.wso?op=NumberToWords>).  
The objective is to ensure the service returns accurate and human-readable English text for given positive numerical inputs, ensuring correctness, robustness, and usability for developers and application integrators.

**2. Scope**

The scope of this test plan includes:

* Features to be tested:
  + Proper handling of valid numerical inputs (from 0 to quadrillions).
  + Validation of incorrect/negative inputs.
  + Compliance with SOAP standards and expected XML structure.
  + Proper error handling and response codes (e.g., 200 OK, 400 Bad Request).
  + Functional and boundary test coverage.
* Types of testing:
  + Manual testing
  + Automated functional testing (via SOAP UI/Postman automation or equivalent)
  + Performance testing (response times, handling of large payloads)
  + Basic accessibility and error messaging verification
* Environments:
  + Browsers (for accessing service docs): Chrome, Firefox, Edge
  + OS: Windows 10, macOS, Linux
  + Tools: Postman, SOAP UI, or any REST/SOAP client
  + Devices: Desktop/laptop (API is service-layer, device-independent)
* Evaluation Criteria:
  + Accuracy of number-to-word conversion
  + Pass/fail rates of test cases
  + Number of defects logged/resolved
  + Consistency of response format
  + Load handling and response times
* Team Roles:
  + QA Lead: Test planning, strategy, and reporting
  + QA Engineer: Test execution, defect logging, automation scripting

**3. Inclusions**

* Introduction to SOAP testing for the service
* Test objectives:
  + Validate correctness of conversion logic
  + Ensure proper response formatting
  + Identify functional or formatting defects
  + Establish baseline performance

**4. Exclusions**

* UI testing (service is API-based only)
* Localization testing (response is always in English)
* Authentication (service is public and open)

**5. Test Environments**

* OS: Windows 10, macOS Ventura, Ubuntu 22.04
* Browsers (only for documentation access): Chrome 124, Firefox 125, Edge 123
* Devices: Laptops/desktops
* Network: Wi-Fi (min 20 Mbps)
* Requirements:
  + Tools like Postman/SOAP UI installed
  + XML parsing plugins
* Access: Open access URL, no credentials required

**6. Defect Reporting Procedure**

* Criteria:
  + Incorrect word conversion
  + Unexpected errors (e.g., null response, missing tags)
  + Deviation from expected SOAP envelope structure
* Reporting steps:
  + Capture input payload
  + Log request/response
  + Attach screenshots and logs
  + Reproduce steps and actual vs expected output
* Triage:
  + Severity 1: Incorrect result
  + Severity 2: Correct but misformatted response
  + Severity 3: Minor UI/doc issues
* Tool: JIRA
* Roles: QA logs; QA Lead reviews; Developer addresses
* Communication: Daily stand-up or Slack/email thread
* Metrics:
  + Defect density
  + Avg. time to resolve
  + % test cases passed

**7. Test Strategy**

**Step 1: Test Case Creation**

* Techniques:
  + Boundary Value Analysis (0, 1, 10, 999, 1,000,000, etc.)
  + Equivalence Class Partitioning (small, medium, large numbers)
  + Use Case Testing: User scenarios like invoice totals, bank balances
  + Error Guessing: Negative numbers, alphanumeric, empty payload

**Step 2: Testing Procedure**

* Smoke Testing: Ensure endpoint is accessible
* Functional Testing: Systematic testing of number ranges
* Load Testing: Stress with high-value inputs
* Defect Logging: Daily triage and logs in JIRA

Types of Testing:

* Smoke, Sanity, Regression
* Positive and Negative Testing
* XML Schema Validation
* Performance (response time and throughput)

**Step 3: Best Practices**

* Shift Left: Early test case planning
* Exploratory Testing for edge cases
* Reuse of test data in automation
* End-to-End: Full SOAP request validation

**8. Test Schedule**

| **Task** | **Start Date** | **End Date** | **Duration** |
| --- | --- | --- | --- |
| Test Plan Creation | May 1, 2025 | May 2, 2025 | 2 Days |
| Test Case Design | May 2, 2025 | May 4, 2025 | 3 Days |
| Environment Setup | May 4, 2025 | May 5, 2025 | 2 Days |
| Test Execution | May 5, 2025 | May 13, 2025 | 9 Days |
| Reporting & Closure | May 13, 2025 | May 14, 2025 | 2 Days |

**9. Test Deliverables**

* Test Plan Document
* Test Scenarios and Test Cases
* Execution Report
* Defect Report
* Test Summary Report

**10. Entry and Exit Criteria**

* Requirement Analysis:
  + Entry: Access to WSDL and service documentation
  + Exit: Confirmed understanding and test data prepared
* Test Execution:
  + Entry: Signed-off test cases and working endpoint
  + Exit: Test results logged, defects reported
* Closure:
  + Entry: All test cases executed
  + Exit: Final report delivered, approval from QA Lead

**11. Tools**

* JIRA (bug tracking)
* Postman/SOAP UI (SOAP test execution)
* Notepad++ / XML viewers
* Excel for test case documentation
* Snipping Tool for screenshots

**12. Risks and Mitigations**

| **Risk** | **Mitigation Strategy** |
| --- | --- |
| Limited team resources | Prioritize critical scenarios |
| Service downtime or URL failure | Notify client, use mock/test endpoints |
| Misinterpretation of requirements | Regular sync with dev/PO for clarification |

**13. Approvals**

Documents to be shared for approval:

* Test Plan
* Test Scenarios & Cases
* Execution and Summary Reports

Approvers: QA Lead, Project Manager, Client Representative