Objective

The objective of this SOAP API test plan is to validate the ISBN10 number using the **IsValidISBN10** API. The service accepts a 10-digit ISBN number (or 9 digits + `X`) via XML POST request and returns a boolean response — `true` if the ISBN is valid, otherwise `false`.

Scope

Scope of Test Plan for the ISBN10 Validation SOAP API:

1. **Functional Testing**
    - Validate correctness of API response for valid and invalid ISBN numbers.
    - Ensure response adheres to expected XML schema and data types.
2. **Data Validation Testing**
    - Validate ISBN input constraints and formats.
    - Test correct and incorrect ISBN values (length, invalid characters, edge cases).
3. **Error Handling Testing**
    - Validate error messages for malformed or missing parameters.
    - Ensure API responds with correct HTTP status codes.
4. **Security Testing**
    - Verify secure access if authentication is needed.
    - Check for input-based vulnerabilities.
5. **Integration Testing**
    - Validate API response when integrated with other services.
6. **Compatibility Testing**
    - Validate API requests on different platforms/tools (Postman, SOAP UI, custom client).

7. **Documentation Review**
    - Ensure API documentation matches actual behavior.

Inclusions

- Send SOAP POST requests with multiple ISBN values.
- Validate API's ability to differentiate valid and invalid ISBN numbers.
- Include edge cases, boundary values, and malformed inputs.

Examples Covered

- Valid ISBN (e.g., `9992158107`)
- ISBN ending in X (e.g., `123456789X`)
- Invalid ISBNs (e.g., short, long, alphanumeric, blank)
- Boundary & special character testing

Test Environments

Component	Description
API URL	`http://webservices.daehosting.com/services/isbnservice.wso`
OS	Windows 10, macOS
Browsers	Chrome, Firefox, Edge, Safari
Devices	Desktop, Mobile (for SOAP Tools)
Tools Used	SOAP UI, Postman, XML Editor
Protocol	SOAP 1.1

Response Format	XML

Defect Reporting Procedure

- **Tool**: JIRA
- **Template**: Title, steps to reproduce, expected vs actual result, screenshots/logs
- **POCs**:
    - Backend: Web Service Dev Team
    - QA Lead: [Your Name]
- **Severity & Priority**: Defined during triage meetings

Test Strategy

1. **Test Case Design**
    - Techniques: Equivalence Partitioning & Boundary Value Analysis
    - Both positive & negative cases included
2. **Execution Flow**
    - Smoke test API endpoint availability
    - Execute functional, boundary, negative, and exploratory tests
3. **Types of Testing**
    - Functional, Regression, Usability, Exploratory
4. **Best Practices**
    - Context-driven, shift-left testing
    - Validate end-to-end flows with sample requests

Test Schedule

Task	Duration
Test Plan Creation	Day 1
Test Case Preparation	Day 2
Execution & Defects	Day 3 – Day 4
Summary Report	Day 5
Sprint Duration	1 Week Total

Test Deliverables

- Test Plan Document
- Test Case Document (Excel or Test Management Tool)
- Execution Logs & Screenshots
- Defect Reports
- Final Summary Report

Entry and Exit Criteria

Requirement Analysis

- Entry: Access to WSDL & endpoint details
- Exit: Requirements fully documented

Test Execution

- Entry: Test cases approved, API reachable
- Exit: Test logs & defect reports submitted

Test Closure

- Entry: Final test results compiled
- Exit: Summary report shared

Tools

- JIRA (Bug Tracking)
- SOAP UI / Postman (Test Execution)
- Excel / Google Sheets (Test Cases)
- Screenshot Tools

Risks and Mitigations

Risk	Mitigation
API service downtime	Use mock/stub server or retry mechanism
Lack of test data	Generate synthetic ISBN data
Limited access to API	Coordinate with service owner early

Approvals

Team will submit the following for stakeholder approval:

- Test Plan
- Test Scenarios & Cases
- Execution Reports
- Final Summary

Testing will proceed to the next phase only after approvals.