**Name**: Mrunal Solanki

# **Test Plan for Number Conversion Service API**

## **1. Objective**

This document outlines the test plan for the **Number Conversion Service API**. The objective is to ensure that all features and functionalities work as expected for users who input any number and receive its corresponding word representation as output.

## **2. Scope**

The scope of this test plan includes:

* **Features to be tested:**
  + Conversion of numbers to words (e.g., 123 → "One Hundred Twenty-Three").
  + Handling of large numbers.
  + Support for various number formats (integers, decimals, negative numbers).
  + Error handling for invalid inputs (e.g., alphabets, special characters).
* **Types of testing:**
  + Manual testing
  + Automated testing
  + Performance testing
  + Accessibility testing
* **Environments:**
  + Various browsers (Google Chrome, Mozilla Firefox, Microsoft Edge).
  + Different operating systems (Windows 10, macOS, Linux).
  + Device types (desktop, laptop, tablet, smartphone).
* **Evaluation criteria:**
  + Number of defects found.
  + Time taken to complete testing.
  + User satisfaction ratings.
* **Team roles and responsibilities:**
  + Test Lead: Oversees testing efforts.
  + Testers: Execute test cases and log defects.
  + Developers: Fix defects and collaborate on issue resolution.

## **3. Inclusions**

* **Introduction:** Overview of the test plan, including purpose, scope, and goals.
* **Test Objectives:**
  + Identify defects in the API.
  + Ensure the API provides accurate number-to-word conversion.
  + Validate API performance under load.
  + Enhance user experience through rigorous testing.

## **4. Exclusions**

* Features related to multi-language number conversion (if not implemented yet).
* API integration with third-party services (if out of scope).
* Advanced mathematical computations beyond number-to-word conversion.

## **5. Test Environments**

* **Operating Systems:** Windows 10, macOS, Linux.
* **Browsers:** Google Chrome, Mozilla Firefox, Microsoft Edge.
* **Devices:** Desktop computers, laptops, tablets, smartphones.
* **Network Connectivity:** Wi-Fi, cellular, wired connections.
* **Hardware/Software Requirements:**
  + Minimum processor: Intel i5 or equivalent.
  + RAM: 8GB or higher.
  + Storage: 10GB free space.
* **Security Protocols:** HTTPS, token-based authentication (if applicable).
* **Access Permissions:** Roles for team members such as testers, developers, and stakeholders.

## **6. Defect Reporting Procedure**

* **Criteria for Identifying Defects:**
  + Incorrect number-to-word conversion.
  + API returns errors for valid inputs.
  + Performance degradation under high load.
* **Steps for Reporting Defects:**
  + Use a designated defect template.
  + Provide detailed reproduction steps.
  + Attach screenshots or logs.
* **Triage and Prioritization:**
  + Assign severity and priority levels.
  + Assign defects to the responsible team members.
* **Tracking Tools:** JIRA, Bugzilla, or any defect tracking tool.
* **Roles and Responsibilities:** Testers, developers, test lead.
* **Communication Channels:** Email, Slack, project management tools.
* **Metrics:**
  + Number of defects found.
  + Time taken to resolve defects.
  + Percentage of defects fixed.

## **7. Test Strategy**

### **Step 1: Test Scenarios and Test Cases Creation**

* Techniques Used:
  + Equivalence Class Partitioning
  + Boundary Value Analysis
  + Decision Table Testing
  + State Transition Testing
  + Use Case Testing
* Additional Methods:
  + Error Guessing
  + Exploratory Testing

### **Step 2: Testing Procedure**

* **Smoke Testing:** Verify API's core functionalities.
* **In-depth Testing:** Execute test cases after stable build passes smoke testing.
* **Multiple Environments Testing:** Ensure compatibility across different platforms.
* **Defect Reporting:** Log defects in tracking tools with daily status updates.
* **Types of Testing:**
  + Smoke Testing
  + Sanity Testing
  + Regression Testing
  + Retesting
  + Usability Testing
  + Functionality & UI Testing (if applicable)

### **Step 3: Best Practices**

* **Context-Driven Testing:** Adjust testing approach based on API behavior.
* **Shift-Left Testing:** Begin testing in the early development stages.
* **Exploratory Testing:** Conduct tests outside predefined test cases.
* **End-to-End Flow Testing:** Simulate complete user interactions.

## **8. Test Schedule**

| **Task** | **Duration** |
| --- | --- |
| Test Plan Creation | 2 Days |
| Test Case Creation | 3 Days |
| Test Execution | 5 Days |
| Defect Reporting | Ongoing |
| Test Summary Report Submission | 2 Days |

## **9. Test Deliverables**

* **Entry and Exit Criteria:** Defined for each testing phase.

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

#### **Requirement Analysis:**

* **Entry:** Receive API documentation.
* **Exit:** Understand and clarify requirements.

#### **Test Execution:**

* **Entry:** Signed-off test scenarios and test cases; application ready for testing.
* **Exit:** Test case reports, defect reports ready.

#### **Test Closure:**

* **Entry:** Test execution completed.
* **Exit:** Test summary reports delivered.

## **11. Tools**

* **Defect Tracking:** JIRA, Bugzilla.
* **Test Case Management:** TestRail, Excel.
* **Performance Testing:** JMeter.
* **API Testing:** Postman, SoapUI.
* **Documentation:** Confluence, Word, Excel.

## **12. Risks and Mitigations**

| Risk | Mitigation Strategy |
| --- | --- |
| Non-availability of resources | Backup resource planning |
| Build URL not working | Collaborate with Dev team for quick resolution |
| Less time for testing | Prioritize critical test cases, conduct risk-based testing |

## **13. Approvals**

* Documents for Client Approval:
  + Test Plan
  + Test Scenarios
  + Test Cases
  + Test Reports

This test plan ensures a systematic approach to testing the **Number Conversion Service API**, covering functional, performance, and usability aspects to deliver a high-quality product.