**Test plan for**

**<<Google Translate>>**

*ChangeLog*

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Change Date** | **By** | **Description** |
| version number | Date of Change | Name of person who made changes | Description of the changes made |
| 001 | 29/04/29 | Vikum Galappaththi |  |
|  |  |  |  |

1 Introduction 2

1.1 Scope 2

1.1.1 In Scope 2

1.1.2 Out of Scope 2

1.2 Quality Objective 2

1.3 Roles and Responsibilities 3

2 Test Methodology 3

2.1 Overview 3

2.2 Test Levels 4

2.3 Bug Triage 4

2.4 Suspension Criteria and Resumption Requirements 4

2.5 Test Completeness 4

3 Test Deliverables 4

4 Resource & Environment Needs 5

4.1 Testing Tools 5

4.2 Test Environment 5

5. TIMEFRAME................................................................................................................................................6

6.RISKMITIGATION........................................................................................................................................7

# Introduction

The Test Plan is designed to prescribe the scope, approach, resources, and schedule of all testing activities of the project Google translate.

The plan identify the items to be tested, the features to be tested, the types of testing to be performed, the personnel responsible for testing, the resources and schedule required to complete testing, and the risks associated with the plan.

## Scope

### In Scope

testing will focus on translation accuracy, language detection, user interface responsiveness, and supported languages.

The target platform for testing is the Google Translate website: https://translate.google.com/

### Out of Scope

**Advanced Features**: Testing of advanced features such as voice translation, handwriting input, camera translation, and conversation mode is out of scope.

**Non-Web Platforms**: Testing of Google Translate mobile applications for Android and iOS devices, as well as standalone applications for desktop, is not included in this testing.

**Integration Testing**: Testing of integration with other Google services or third-party applications is not part of this QA effort.

**Security Testing:** In-depth security testing, including penetration testing and vulnerability assessment, is not covered in this testing.

**Browser Plugins/Add-ons:** Testing of browser plugins or add-ons for Google Translate, if any, is not included.

## Quality Objective

To ensure the accuracy, reliability, and usability of the Google Translate online product for translating sentences/words between Sinhala, Tamil, and English languages.

Objectives for the project

* Enhance Translation Accuracy
* Expand Language Support
* Optimize User Experience
* Improve Language Detection
* Enable Cross-Platform Integration
* Enhance Accessibility

## Roles and Responsibilities

Detail description of the Roles and responsibilities of different team members like

* Test Manager - Manage the whole project

-Define project directions

-Acquire appropriate resources

* Test Administrator - Builds up and ensures test environment and assets are managed and maintained

-Support Tester to use the test environment for test execution

* Developers - Implement the unit test cases
* SQA Team - Take in charge of quality assurance

-Check to confirm whether the testing process is meeting specified requirements

# Test Methodology

## Overview

Test Mythology: Agile Testing for Google Translate

**Sprint Planning:**

Collaborate with the development team and stakeholders to understand the upcoming features and changes to Google Translate.

Identify high-priority user stories related to translation accuracy, language detection, UI responsiveness, and browser compatibility.

Break down user stories into smaller, testable tasks and estimate effort for each task.

**Daily Standups:**

Participate in daily standup meetings to discuss progress, any roadblocks, and plan testing activities for the day.

Provide updates on completed testing tasks, any issues encountered, and planned testing for the current sprint.

**Continuous Testing:**

Begin testing as soon as development work is completed on a feature or user story.

Perform exploratory testing to uncover any unexpected issues or edge cases.

Automate repetitive test cases and regression tests to ensure consistent coverage across sprints.

**Collaboration and Communication:**

Maintain open communication with developers, product owners, and other stakeholders throughout the sprint.

## Test Levels

1. **Unit Testing -** Validate the functionality of individual components or units of code.
2. **Integration Testing -** Verify the interactions and integration between different modules or components of the Google Translate system.
3. **System Testing -**Evaluate the behavior and functionality of the Google Translate system as a whole.
4. **Acceptance Testing -**Determine whether the Google Translate system meets the specified requirements and satisfies user needs.
5. **Regression Testing -**Ensure that new changes or updates to the Google Translate system do not introduce regressions or unintended side effects.

## Bug Triage

The goal of the triage is to

* To define the type of resolution for each bug
* To prioritize bugs and determine a schedule for all “To Be Fixed Bugs’.

## Suspension Criteria and Resumption Requirements

If the team members report that there are 40% of test cases failed, suspend testing until the development team fixes all the failed cases.

## Test Completeness

Following testing criteria will be completed

* 100% test coverage
* All Manual & Automated Test cases executed
* All open bugs are fixed or will be fixed in next release

# Test Deliverables

Here are the test deliverables

|  |
| --- |
| * Test Plan * Test Cases * Requirement Traceability Matrix * Bug Reports * Test Strategy * Test Metrics * Customer Sign Off |

# Resource & Environment Needs

## Testing Tools

Here are the tools

* Jira
* Selenium with Java
* Browserstack

Required to test the project

## Test Environment

Here are the test Environment requirements that will be used to test the Application.

**Operating Systems:**

1. Windows
2. macOS
3. Linux

**Web Browsers:**

1.Google Chrome

2.Mozilla Firefox

3.Safari

4.Microsoft Edge

**Devices:**

1.Desktop computers

2.Laptops

3.Tablets

4.Smartphones

**Screen Resolutions:**

Various screen resolutions ranging from desktop monitors to mobile devices.

**Network Conditions:**

Testing under different network conditions, including high-speed broadband, 3G, and 4G, to ensure optimal performance across various network environments.

**Localization:**

Testing with different language settings and locales to ensure accurate translation and language detection capabilities. **input Devices:**

Keyboard and mouse or touchpad for input interaction with the application.

# 5 Timeframe

The timeframe for the Google Translate project can vary depending on factors such as project scope, complexity, resources, and specific requirements. However, here's a general timeframe that could be considered for a typical iteration of the project:

**1.Requirements Gathering and Analysis: 1 week**

Quickly gather and analyze high-level requirements, focusing on essential features and functionalities. Prioritize language support and translation accuracy as primary objectives.

1. **Design and Planning: 1 week**

Develop a simplified test plan outlining key testing objectives, test cases, and criteria. Coordinate closely with stakeholders to finalize project scope and priorities.

**3.Development: 2 weeks**

Rapidly implement core features and functionalities, focusing on language support, translation accuracy, and user interface improvements. Utilize agile development methodologies to iterate quickly and adapt to changing requirements.

**4.Testing: 1 week**

Conduct focused testing, including unit testing, integration testing, and system testing. Prioritize critical test scenarios and functionalities to ensure basic functionality and stability.

**5.Bug Fixing and Optimization: 1 week**

Address critical bugs and issues identified during testing. Optimize performance, security, and usability aspects of the application to meet minimum quality standards.

Documentation and Reporting: 1 week

Document key testing activities, test results, and bug fixes. Prepare a concise test report summarizing findings and communicating project status to stakeholders

**Proposed Schedule:**

|  |  |
| --- | --- |
| Week 1 | Requirements Gathering and Analysis |
| Week 2 | Design and Planning |
| Week 3-4 | Development  Testing  Bug Fixing and Optimization  Documentation and Reporting |

**6 Risk mitigation**

1. **Translation Accuracy Issues:**

Risk: Inaccurate translations could lead to misunderstandings or miscommunication.

Mitigation: Implement rigorous testing procedures to validate translation accuracy. Utilize automated testing tools, linguistic experts, and crowd-sourced feedback to continuously improve translation algorithms and language models.

1. **Language Support Limitations:**

Risk: Limited language support may exclude users from certain regions or linguistic backgrounds.

Mitigation: Prioritize the addition of languages based on user demand and market analysis. Continuously expand language support through agile development methodologies and regular updates.

**3.Performance Degradation Under High Load:**

Risk: Performance issues during peak usage periods could result in slow response times or service interruptions.

Mitigation: Conduct thorough performance testing to identify bottlenecks and optimize system resources. Implement load balancing, caching mechanisms, and scaling strategies to handle increased traffic efficiently.

**4.Security Vulnerabilities:**

Risk: Security breaches or data leaks could compromise user privacy and trust in the service.

Mitigation: Implement robust security measures such as data encryption, access controls, and regular security audits. Stay informed about emerging security threats and promptly patch vulnerabilities.

**5.Compatibility Issues Across Platforms:**

Risk: Incompatibility with certain browsers, devices, or operating systems may result in a suboptimal user experience.

Mitigation: Perform extensive cross-browser and cross-platform testing to ensure compatibility. Utilize responsive design principles and progressive enhancement techniques to adapt to different screen sizes and resolutions.

**6.Regulatory Compliance Challenges:**

Risk: Failure to comply with regulatory requirements, such as data protection laws, could lead to legal consequences or reputational damage.

Mitigation: Stay informed about relevant regulations and ensure compliance through proactive measures such as data anonymization, consent management, and transparency in data handling practices.