Test Plan

Google Translate

V1.0
17/07/2024

Test Plan Change Log

Identifier	*Version*	*Changed Date*	*Changed By*	*Description*
Test_Plan_Google_Translate	V1.0	17/07/2024	Sagarika.W.	Initial test plan

WireApps

Table of Contents

1 Introduction

Google Translate (https://translate.google.com/) is an online service provided by Google that translates text, images, documents and websites from one language to another. This QA process aims to ensure that the translation between Sinhala, Tamil and English is accurate, reliable and user-friendly.

In Scope:

- Translation accuracy between Sinhala, Tamil, and English.
- Functionality of key features (text input, text output speech-to-text, and text-to-speech).
- Usability and user interface assessment.

Out of Scope:

In this test plan, we are considering only the translation of texts, not images, documents, or websites. Additionally, we are not considering the 'Google Translate' mobile app.

1.1 Objective

- Verify the accuracy of translations between Sinhala, Tamil, and English.
- Assess the user interface and user experience.
- Identify any bugs or issues in the translation process.

1.2 Test Environmnet

- Browser: Google Chrome/Microsoft Edge/Mozilla Firefox
- Device: Desktop/Laptop
- URL: https://translate.google.com/

1.3 Roles and Responsibilities

Role	***Team Member***	***Responsibilities***
QA Manager		Acts as the head of the team. Get the daily update regarding the procedure maintained.
QA Engineer	Sagarika.W.	Create test plan. Create test cases Identify test data. Execute test conditions and mark-off result. Prepare software bug report. Prepare test summary report.

2 Test Approch

2.1 Methodology

The system testing is continuing with the Agile/Scrum Model. The Software Testing Life Cycle (STLC) in Agile defines a series of activities conducted to perform Software Testing. It identifies what test activities to carry out and when to accomplish those test activities. In the Agile STLC process, each activity is carried out iteratively and incrementally, aligning with the sprints. Each sprint has different goals and deliverables, ensuring continuous improvement and adaptability throughout the development cycle.

2.2 Test Levels

These testing types will be completed after identifying the areas of the application to be tested. The types of tests related to Black Box testing are considered as follows:

Functional Tests

Functional testing focuses on the functional requirements of the software and is performed to confirm that the application operates accurately according to the expected specifications and requirements, ensuring that interfaces to external systems are properly working. We will test the main functions of the application by checking the following criteria:

- Use sample texts in Sinhala, Tamil, and English for translation.
- Test core functionalities like text input, text output, copying translated text, and switching between languages.
- Perform both positive and negative testing to ensure robustness.

Compatibility Tests

Compatibility testing involves verifying that the functions of the application work correctly across multiple browsers. The QA team will perform compatibility tests for the web app using Google Chrome, Microsoft Edge, and Mozilla Firefox on the Windows 11 platform.

Interface Testing

Interface testing will be performed to ensure that the product works as expected in the way a typical user would interact with it.

Stress & Performance Tests (Optional)

As a future plan, we are planning to introduce stress and performance testing using Apache JMeter.

Automated Testing (Optional)

As a future plan, web automation using Cypress can be implemented for the identified scenarios.

2.3 Suspension Criteria and Resumption Requirements

N/A

2.4 Test Completeness

For instance, a few criteria to check Test Completeness would be,

- 100% test case update
- All manual test cases executed
- All open bugs are fixed or will be fixed in future sprint
- All the required documents are completed before the sign-off

3 Test Deliverables

Following documents will be delivered during different phases of the testing lifecycle

- Test plan
- Test cases
- Test summary report
- Bug report

4 Resource & Environmental Needs

4.1 Testing Tools

Process	***Tool***
Test case create/update	Excel (will be moved to TestLink in future)
Test case execution	Excel (will be moved to TestLink in future)
Defect tracking	Excel (will be moved Jira in future)
Test summary report	Excel/PDF/Word

4.2 Test Environment

Following hardware & softwares are required in order to test.

- Devices - Desktops/Laptops/Mobiles
- Browsers – Google Chrome/Microsoft Edge/Mozilla Firefox
- Connectivity - Internet

5 Entry and Exit Criteria

5.1 Entry Criteria

- All test hardware and software must be successfully installed and functioning properly.
- All necessary documentation, and requirements information should be available to allow testers to perform the tests and judge correct behavior.
- All standard software tools, including testing tools, must be successfully installed and functioning properly.
- Proper test data must be available.
- Test scenarios and test cases must be peer-reviewed.

Entry Criteria	***Status***
Windows desktop/laptop	Available
Internet connectivity	Available
Google Chrome/Microsoft Edge/Mozilla Firefox	Available
Mobile phone with Google Chrome	Available

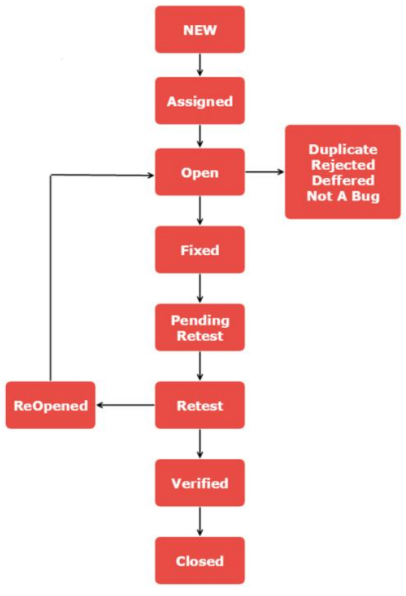
5.2 Exit Criteria

- Functional testing should be completed.
- Relevant bugs should be reported.

6 Test Strategy

6.1 Bug Life Cycle

Following the bug life cycle process for defect tracking for Google Translate, the QA team can immediately start the test plan executions. The QA team identifies issues while executing the test plan, which are then tracked in Excel and reported to the Google team. After the Google team resolves all the identified issues and release a new version, QA team will performs verification, and provide QA sign-off.



6.2 Bug Severity and Priority

Bug Severity

A severity level is supplied for each bug identified.

Severity ID	***Severity***	***Severity description***
1	Critical	A defect that completely hampers or blocks testing of the product/feature is a critical defect. If the application crashes or it becomes unusable/not able to proceed further, the defect could be classified under critical severity. Eg: After typing the correct text, the application crashes or throws the error message.
2	Major	The application that is not meeting its requirements/use case(s) and behaves differently than expected, it can be classified under Major Severity. Eg: When user enter the text and returns completely incorrect translation.
3	Minor	A defect occurs when the application doesn't meet certain criteria or still exhibits some unnatural behavior, but the impact is negligible to some extent, or it doesn't have a major impact on the application. Eg: The sentence is translated incorrectly but the meaning is correct.
4	Low	Any cosmetic defects including alignment issues, or font format can be classified under Low Severity. Eg: If there are any issues that the text areas are not aligned well.

6.3 Bug Priority Definition

Priority Level	***Priority***	***Priority Description***
Priority #1	(High)	The defect must be resolved as soon as possible as it affects the system severely and cannot be used until it is fixed. Eg: The issues that translates to the incorrect languages than expected language.
Priority #2	(Medium)	A defect with this priority must be in contention to be fixed as it could also deal with functionality issues which are not as per expectation. During the normal course of development activities, the defect should be resolved. Eg: The translations do not translate the entire sentence but gives the correct meaning
Priority #3	(Low)	The defect is an irritant which should be repaired, but repair can be deferred until after more serious defects have been fixed. Eg: If there are any issues that the text areas are not aligned well.

7 Test Schedule

Task ID	*Name*	*Start date*	*End Date*	*Effort By*	*Comment*
001	SRS Received	16/07/2024			
002	Begin Test Planning	16/07/2024	16/07/2024	Sagarika.W.	
003	Review Test Plan	16/07/2024			
004	Getting the Approval	N/A			
005	Creating Test Cases	17/07/2024	17/07/2024	Sagarika.W.	
006	Test Case Execution, bugs report	17/07/2024	17/07/2024	Sagarika.W.	
007	Test Summary Report	17/07/2024	17/07/2024	Sagarika.W.	
008	Release Documents	18/07/2024	18/07/2024	Sagarika.W.	

7.1 Measurement Plan

Task ID	*Date*	*Status*	*Comment*
002	16/07/2024	Completed	
003	16/07/2024	Completed	
005	17/07/2024	Completed	
006	17/07/2024	Completed	
007	17/07/2024	Completed	
008	18/07/2024	Completed	

8 Risks and Contingencies

- Scope creep (last-minute addition of new requirements) impacts deadlines. Eg: Requesting to test for additional languages or requesting to test mobile application
- Any downtime of the test system will significantly impact the testing cycle
- Reported Bugs will not be fixed on scheduled time periods

9 Approvals

Party	*Approval (Yes/No)*	*Approved Date*	*Comment*
QA Manager			
QA Lead			

10 Terms/Acronyms

Term/Acronym	*Definition*
Mobile app	Google translate mobile application
Application	Google translate web application