**Test Plan Document**

**For**

**Google Translation App**

**Table of Contents**

[**1** Introduction 3](#_Toc164678528)

[1.1 Overview 3](#_Toc164678529)

[1.1.1 Project 3](#_Toc164678530)

[1.1.2 Document Overview 3](#_Toc164678531)

[1.2 Purpose 3](#_Toc164678532)

[1.3 Scope 3](#_Toc164678533)

[1.3.1 In scope 3](#_Toc164678534)

[1.3.2 Out of scope 3](#_Toc164678535)

[1.3.3 Functional Requirements 4](#_Toc164678536)

[1.4 Risk and Dependencies 4](#_Toc164678537)

[1.4.1 Product Risk 4](#_Toc164678538)

[1.4.2 Project Risk 5](#_Toc164678539)

[2 Test 5](#_Toc164678540)

[2.1 Test Deliverables & Schedule 6](#_Toc164678541)

[2.2 Test Design 6](#_Toc164678542)

[2.3 Test Execution 7](#_Toc164678543)

[2.3.1 Entry Criteria 7](#_Toc164678544)

[2.3.2 Exit Criteria 7](#_Toc164678545)

[2.3.3 On Hold Criteria 7](#_Toc164678546)

[2.3.4 Test Data Requirements 7](#_Toc164678547)

[2.3.5 Test Metrics 7](#_Toc164678548)

[2.3.6 Test Report 8](#_Toc164678549)

[2.3.7 Defect Management 8](#_Toc164678550)

[3 Resource 8](#_Toc164678551)

[3.1 QA Team 8](#_Toc164678552)

[3.2 Team/Triage Meetings 8](#_Toc164678553)

[3.3 Test Environments 8](#_Toc164678554)

[3.4 Training/Tools Required 8](#_Toc164678555)

[4 Assumptions 9](#_Toc164678556)

# **1** Introduction

## Overview

### Project

The google translation app is, free translation app where a user can go and translate any language test to the desired language they understand

### 1.1.2 Document Overview

This document describes the scope of the overall test effort and provides a record of the test planning process for. Typically, this document mentions the requirements, risks, test approach, test deliverables, and quality objectives.

## 1.2 Purpose

Purpose of QATP is to achieve the requirements of Google Translation App in a quality product. The test cycle will be followed according to the guidelines of this document.

## 1.3 Scope

### 1.3.1 In scope

The main scope of the Google Translation application is to perform language traslation activities using the Google Translation App. The system will handle online customers.

* The application persists of the core application, customer application.
* Do language translation
* Select a language and translate the test
* Do test translation but typing a test or copying a text
* Auto capture the language
* Upload a image to be translated
* Upload a document to be translated
* Capture the text from a website and translate

### 1.3.2 Out of scope

* Multi-language availability is out of scope.
* The system must be 99.9% available to perform as required during the period of a mission.(performance testing)
* Multi- lingual functionality test
* Third party payment gateway test

### 1.3.3 Functional Requirements

|  |  |  |
| --- | --- | --- |
| **Feature Name** | **Feature Description** | **Business Priority** |
| Enter Text | User can enter a valid language text | High Risk |
| Auto detect the language | The app automatically detect the language that was entered | High Risk |
| Shoe the transaction on the default language | The text get translated in the selected language | High Risk |
| Change the language to be translated | The user change the language to be translated in to | High Risk |
| Uploads the image to be translated | The user selects or bowser a image to translated | High Risk |
| Image get translated | The document get translated in the selected language | Low Risk |
| Uploads the document to be translated | The user selects or bowser a document to translated | High Risk |
| Document get translated | The image get translated in the selected language | Medium Risk |

## 1.4 Risk and Dependencies

### 1.4.1 Product Risk

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk | Impact | Mitigation | Contingency | Owner |
| Changes to the functionality may negate the tests already written and we may have to rewrite test cases. | It will take more time and delay for the execution | Functionalities of the application must be gathered as much as possible beforehand | Before starting development, fixed the requirements which can’t change in future | PM |
| Non-availability of Independent Test environment and accessibility | Delay for testing | Due to the non-availability of the environment, the schedule gets impacted and will lead to delayed start of Test execution | Before starting testing fixed all environment | QA |
| Because the developer works off site, Weekly delivery may not be possible. | Delay for test execution, close bugs. | Developers should push the changes into the QA environment as soon as they develop | Before taking work off complete all the work or assign someone to complete task within deadline | Developer |

### 1.4.2 Project Risk

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3rd party gateways and API are down | Cannot complete the test flow. | Keep a script/mock to respond | Change the test execution flow where can be tested without 3rd party flow | Dev |
| Non-availability of Independent Test environment and accessibility | High | Due to the non-availability of the environment, the schedule gets impacted and will lead to delayed start of Test execution | A proper test environment should be released to QA without delay | Developer |
| Late delivery From Dev team | Delay for test execution, close bugs. | Developers should push the changes into the QA environment as soon as they develop | Before taking work off complete all the work or assign someone to complete task within deadline | Developer |
| Changes to the functionality may negate the tests already written and we may have to rewrite test cases. | High | Functionalities of the application must be gathered as much as possible beforehand | At the start of the project, proper requirements must be gathered from the BA to begin development | QA |

# 2 Test

## 2.1 Test Deliverables & Schedule

|  |  |
| --- | --- |
| Deliverable | Description |
| Quality Assurance Test Estimate | Estimate gives the total time takes for QA cycle for Google Translation project |
| Quality Assurance Test Plan (QATP) | Test plan is a document detailing the objectives, resources, and processes for a specific test for a software or hardware product(Google Translation project). The plan typically contains a detailed understanding of the eventual workflow. |
| Quality Assurance Test Scenario (QATS) | For the agreed requirements, the test QA team identified is listed in the Google Translation project document and will be executed in the test execution phase. |
| Traceability Matrix | Mapping the finalized requirements with written test scenarios. This will help to confirm all the requirements in SRS are covered by the Google Translation project test scenarios document. |
| Test Results | Test report is an assessment of how well the Testing is performed. Based on the test report, stakeholders can evaluate the quality of the tested product and make a decision on the software (Google Translation project) release. |
| Defect Report | The mismatch from the agreed requirement, improvement expected from user point are logged and submitted as a report after execution. |
| Release Notes | When QA cycle is completed and Google Translation project product is ready for release, QA will give the release note with the release details of Google Translation project |

## 2.2 Test Design

Test is designed based on the requirement mentioned in the wireframe. The functionalities are broken down into small features and test scenarios are created for it. Based on feature severity and priority, test scenarios' priority is also set.

* Understand the Requirements
* Create test plan document
* Create test scenario document
* Review the test scenario document
* Make changes when any requirements mismatch based on the review
* Each functionality will be covered in one/many tests.
* All the scenarios will be ready before execution.

## 2.3 Test Execution

QA will start the test execution after a testable module is released by the Development team and all test scenarios are ready.

* QA will start with Smoke test on Vast release
* QA will do complete Functional teston the features released
* QA will doRegression teston bug fix components
* API & UI Manual testing
* UI & API Automation testing using Katalon

### 2.3.1 Entry Criteria

* Test scope identified
* Test scenarios identified
* Test environment ready
* Independent testable component releases to QA

### 2.3.2 Exit Criteria

* Planned tests are executed
* All agreed requirements are tested
* No open bugs (exception for Trivial severity bugs)

### 2.3.3 On Hold Criteria

* Critical bug/showstopper is found in test execution
* Dependent module is not functioning / Failing

### 2.3.4 Test Data Requirements

* Login credentials from backend to test Admin portal
* Google Translation project Test Data
* API postman collections.

### 2.3.5 Test Metrics

* Test Coverage - Requirement based tests which include functional and nonfunctional tests.
* Test Status - The number of tests planned, ran, passed, failed, or blocked would check the status of testing.
* Defect Status - Number of defects reported against the requirement.

### 2.3.6 Test Report

* Manual Tests are reported in the Test results document.
* API Tests are Reported in the API test results document.
* Load testing report received via mail
* Automation test report received via mail
* Bugs are Updated in Defect report with jira ID

### 2.3.7 Defect Management

* Defects found are reported in JIRA
* Reported defect IDs are tracked in Test Results and Traceability matrix

# 3 Resource

## 3.1 QA Team

|  |  |  |
| --- | --- | --- |
| QA lead | Peter Mariampillai | Check and Manage the QA engineer work |
| Associate QA Engineer |  | Create test plan document,  Prepare test estimate,  Traceability matrix Create test scenario document  Create test execution document  Execute test Test reporting, Defect reporting Release note, Assign Bugs in jira |

## 3.2 Team/Triage Meetings

* Daily stand up meeting
* Weekly demo
* Check point meetings

## 3.3 Test Environments

* Windows Chrome
* Windows Edge
* QA URL
  + https://translate.google.com/?sl=auto&tl=en&op=translate

## 3.4 Training/Tools Required

* Automation tools - Selenium and katalon
* Performance testing tool - JMeter
* API testing - Postman

# 4 Assumptions

* No amendments to functions after signoff test plan and for the sprint
* Third parties web are integrated
* Users who are registered in the system can only access the system to proceed with operations.
* User roles are distinguished based on the job role
* It is assumed that the client will be supportive throughout the development