**Functional Test Report for Inji Verify v 0.9.0**

**05-07-2024**

# 

[Testing Scope 3](#_Toc171361164)

[Test Approach 4](#_Toc171361165)

[Verified configuration 4](#_Toc171361166)

[Feature Health 5](#_Toc171361167)

[Test execution statistics 6](#_Toc171361168)

[Functional test results 6](#_Toc171361169)

[Test results for Upload and Scan 7](#_Toc171361170)

[Detailed Test metrics 8](#_Toc171361171)

# 

# Testing Scope

The scope of testing is to verify fitment to the specification from the perspective of

* Functionality
* Deploy ability
* Configurability
* Customizability

Verification is performed not only from the end user perspective but also from the System Integrator (SI) point of view. Hence Configurability and Extensibility of the software is also assessed. This ensures the readiness of software for use in multiple countries. MOSIP is an “API First” product platform.

Testing scope has been focused on the below features:

* Inji Verify Home page
* Verify Scan Feature
* Verify Upload Feature

Below are the combinations that QA verified and Certified INJI Verify:

**Upload feature Verification:**

1. Windows using Edge, Firefox and Chrome browsers.
2. MAC using Safari browser.

**Scan functionality Verification:**

1. MAC (Laptop) with a front camera of 2 megapixel and Safari browser.
2. Mobile phone Android with camera 16 megapixel and above using browsers Chrome.
3. iPhone with 24-megapixel camera using Safari browser.

**Note:** Unable to verify Scan functionality in windows.

# Test Approach

Persona based approach has been adopted to perform the IV&V, by simulating test scenarios that resemble a real-time implementation.

A Persona is a fictional character/user profile created to represent a user type that might use a product/or a service in a similar way. Persona based testing is a software testing technique that puts software testers in the customer's shoes, assesses their needs from the software and thereby determines use cases/scenarios that the customers will execute. The persona needs may be addressed through any of the following.

* Functionality
* deploy ability
* Configurability
* Customizability

The verification methods may differ based on how the need was addressed.

# Verified configuration

Verification is performed on various configurations as mentioned below

* Default configuration - with 1 Lang
  + English

# Feature Health

# Test execution statistics

## Functional test results

Below are the test metrics by performing functional testing. The process followed was black box testing which based its test cases on the specifications of the software component under test. Functional test was performed in combination of individual module testing as well as integration testing. Test data were prepared in line with the user stories. Expected results were monitored by examining the user interface. The coverage includes GUI testing, System testing, End-To-End flows across multiple configurations. The testing cycle included simulation of multiple identity schema and respective UI schema configurations.

|  |  |  |  |
| --- | --- | --- | --- |
| **Total** | **Passed** | **Failed** | **Skipped** |
| 253 | 198 | 39 | 16 |
| Test Rate: 93%, With Pass Rate: 83% | | | |

Here is the detailed breakdown of metrics for each module:

|  |  |  |
| --- | --- | --- |
|  | | **Test cases** |
| INJI Verify UI Verification | Total | 253 |
| Passed | 198 |
| Failed | 39 |
| Skipped | 16 |

## Test results for Upload and Scan

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Upload button** | **Upload feature** | **Scan Button** | **Scan Feature** | **Overall layout** |
| **Windows** | **Firefox** | Pass | Pass | Pass | In-consistent (Camera resolutions) | Pass |
| **Edge** | Pass | Pass | Pass | In-consistent (Camera resolutions) | Pass |
| **Safari** | NA | NA | NA | NA | NA |
| **Chrome** | Pass | Pass | Pass | In-consistent (Camera resolutions) | Pass |
|  | | | | | | |
| **Mac** | **Firefox** | Pass | Pass | Pass | Pass | Pass |
| **Edge** | Pass | Pass | Pass | Pass | Pass |
| **Safari** | Pass | Pass | Pass | Pass | Pass |
| **Chrome** | Pass | Pass | Pass | Pass | Pass |
|  | | | | | | |
| **iPhone** | **Firefox** | Pass | Pass | Pass | Pass | Pass |
| **Edge** | Pass | Pass | Pass | Pass | Pass |
| **Safari** | Pass | Pass | Pass | Pass | Pass |
| **Chrome** | Pass | Pass | Pass | Pass | Pass |
|  | | | | | | |
| **Android** | **Firefox** | Pass | Pass | Pass | Pass | Pass |
| **Edge** | Pass | Pass | Pass | Pass | Pass |
| **Safari** | NA | NA | NA | NA | NA |
| **Chrome** | Pass | Pass | Pass | Pass | Pass |

## Detailed Test metrics

Below are the detailed test metrics by performing manual/automation testing. The project metrics are derived from Defect density, Test coverage, Test execution coverage, test tracking and efficiency.

The various metrics that assist in test tracking and efficiency are as follows:

* Passed Test Cases Coverage: It measures the percentage of passed test cases. (Number of passed tests / Total number of tests executed) x 100
* Failed Test Case Coverage: It measures the percentage of all the failed test cases. (Number of failed tests / Total number of test cases executed) x 100

Git hub link for the xls file:

<TBD>