**Functional Test Report for**

**eSignet Sign Up Portal v 1.3.0**

**22-02-2024**

# 

[Testing Scope 3](#_heading=h.3znysh7)

[Test Approach 3](#_heading=h.2et92p0)

[Verified configuration 4](#_heading=h.tyjcwt)

[Feature Health 5](#_heading=h.1t3h5sf)

[Test execution statistics 5](#_heading=h.2s8eyo1)

[Functional test results 5](#_heading=h.17dp8vu)

[Detailed Test metrics](#_heading=h.3rdcrjn) 6

Device and Browser details 7

[Sonar Report](#_heading=h.3rdcrjn) 8

# 

# 

# Testing Scope

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

* Functionality
* Deployability
* 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 readiness of software for use in multiple countries. Since MOSIP is an “API First” product platform, Verification scope required comprehensive automation testing for all the MOSIP APIs. An automation Test Rig is created for the same.

# 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
* Deployability
* Configurability
* Customizability

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

For regression check, “MOSIP Test Rig” - an automation testing suite - which is indigenously designed and developed for supporting persona based testing. MOSIP Test Rig covers the end to end test execution and reporting. The end to end functional test scenarios are written starting from pre-registration, to creation of packet in registration center, processing the packet through the registration processor, generating UIN and authenticating identity using IDA through various permutation and combinations of cases being covered. MOSIP Test Rig will be an open source artifact which can also be enhanced and used by countries to validate the SI deliveries before going live. Persona classes include both negative and positive personas. Negative persona classes include users like Bribed Registration Office, Malicious Insider etc. The needs of positive persona classes must be met, whereas the needs of negative persona classes must be effectively restricted by the software.

# Verified configuration

Verification is performed on various configurations as mentioned below

* Default configuration - with 2 Lang (English/Khmer)

**Main feature tested:**

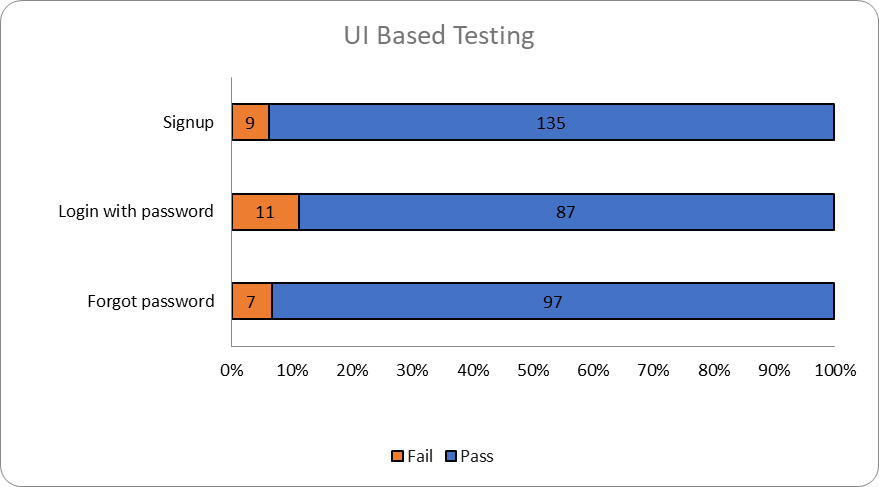
Signup Portal

Login with Password

Forgot Password

# 

# Feature Health



# Test execution statistics

## Functional test results

Below are the test metrics by performing functional testing using mock MDS, mock Auth and mock ABIS. 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 inline 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 languages and configurations. The testing cycle included simulation of multiple identity schema and respective UI schema configurations.

|  |  |  |  |
| --- | --- | --- | --- |
| **Total** | **Passed** | **Failed** | **Skipped** |
| 1604 | 1524 | 76 | 4 |
| Test Rate: 99% with Pass rate: 95% | | | |

Here is the detailed breakdown:

|  |  |  |
| --- | --- | --- |
|  | | **Test cases** |
| API Based Testing | Total | 1256 |
| Passed | 1205 |
| Failed | 49 |
| Skipped | 2 |
| UI Based testing | Total | 348 |
| Passed | 319 |
| Failed | 27 |
| Skipped | 2 |

**Note:** In API Based testing, 2 test cases are marked as skipped as they were not automated.

In UI Based testing, 2 test cases are marked as skipped as they were enhancement test cases.

## 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:

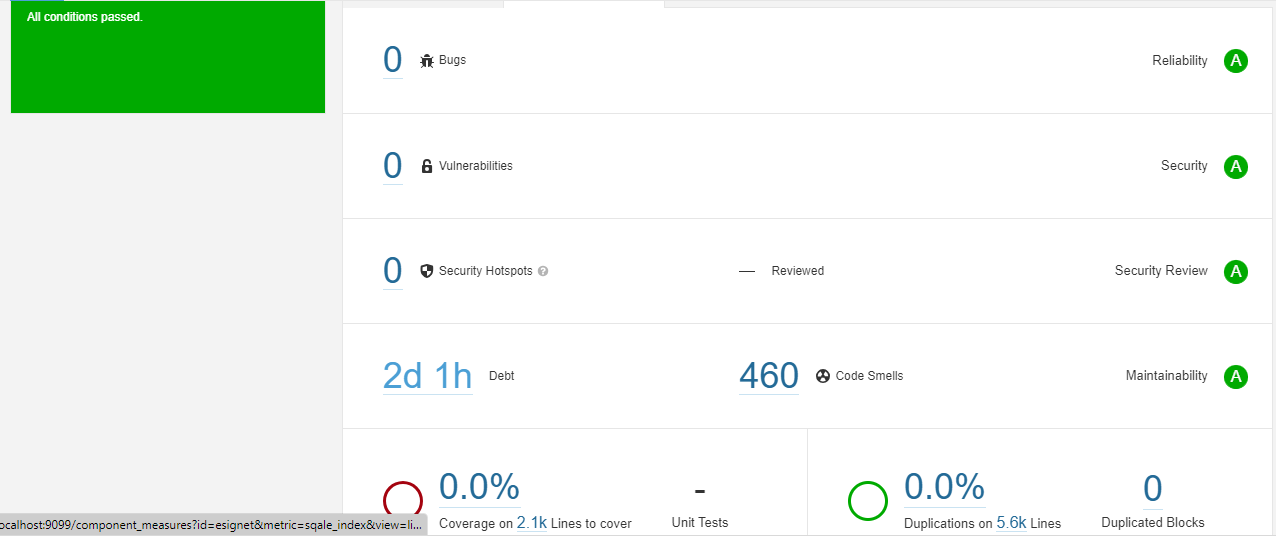
<TBD>;

Device and Browser details:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Mobile specification** | | | | | |
| Device | Browser | Browser version | OS Version | Display resolution | Screen size |
| iphone 15 plus | Safari | 17 | 17.2 | 1290x2796 pixels | 6.7 inches |
| iphone 7 | Safari | 15 | 15.6 | 750 x 1334 pixels | 4.7 inches |
| Vivo y73 | chrome | 120.0.6099.193 | Android 13 | 1080x2400 pixels | 6.44 inches |
| Redmi 7A | chrome | 120.0.6099.145 | Android 10 | 720x1440 pixels | 5.45 inches |
| Samsung Galaxy S5 | chrome | 120.0.6099.145 | Android 4.4.2 | 1080 x 1920 pixels | 5.1 inches |
| Moto G4 | chrome | 120.0.6099.145 | Android 6.0.1 | 1080 x 1920 pixels | 5.5 inches |
| **Desktop browser specification** | | | | | |
| Browser | Browser version | | | | |
| Chrome | 120.0.6099.110 | | | | |
| Mozilla Firefox | 122.0 (20240118164516) | | | | |
| Microsoft Edge | 121.0.2277.128 | | | | |
| Duckduckgo | 0.66.1 | | | | |
| Opera | 107.0.5045.21 | | | | |
| Safari | 16.6 | | | | |

Sonar Report:

eSignet:



eSignet Signup Repo:

