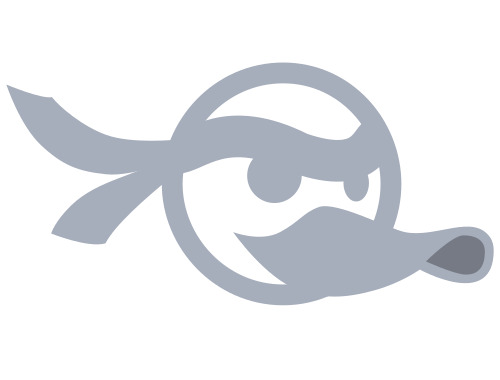
****

HIAAA

System Test Plan

Document Category: TP

Version:<1.1>

Status:<Draft>

Last Updated:<April 14, 2025>

Contents

[Contents 2](#_Toc1725856)

[1 Introduction 4](#_Toc1725857)

[1.1 Purpose 4](#_Toc1725858)

[1.2 Scope 4](#_Toc1725859)

[1.3 Target Audience 4](#_Toc1725860)

[1.4 Related documents 4](#_Toc1725861)

[1.5 Glossary 5](#_Toc1725862)

[2 Setup and Test environment 6](#_Toc1725863)

[3 Tests 7](#_Toc1725864)

[3.1.1 Functional Tests 7](#_Toc1725865)

[3.1.2 Usability Tests 7](#_Toc1725866)

[3.1.3 Business Cycle Tests 8](#_Toc1725867)

[4 Non-Functional Tests 9](#_Toc1725868)

[4.1.1 Scale Tests 9](#_Toc1725869)

[4.1.2 Performance Tests 9](#_Toc1725870)

[4.1.3 Security Tests 9](#_Toc1725871)

[4.1.4 Compatibility Tests 10](#_Toc1725872)

[Approvals 11](#_Toc1725873)

[History 12](#_Toc1725874)

# Introduction

## Purpose

The purpose of this document is to describe the system test plan for <system-name> and will specifically cover the testing of functional and non-functional requirements.

## Scope

*[Use this place to describe the big picture and where this component would fit in the overall architecture. Provide references & pointers to additional architecture & relevant docs to lead the user to get more details on the overall architecture. Example: more detailed information on the architecture can be found in ArchDoc1 mentioned under section 1.4]*

*[Depending on the context and the actual run book component, This could differ significantly but the operators should know what component he is operating on and it’s place in the big picture and the basic message flow.*

*Example: if the run book is for a certain custom bridge or cache, it is desirable for the operator to know where the messages originate, how they access the service on this run book and what they do with the service response. This should be described here to the operator or provide reference docs from where he can get this info]*

*[In some cases however, it may not possible to provide a detailed end-end diagram.. The idea is always provide the operator with the context what he is operating on and how it fits in the big picture.]*

*[Provide high level introduction of the <component> deployment model here. Whether it’s deployed in High available/FT mode, how many instances of the component are deployed in the cluster etc. Having a diagram to explain the deployment model is extremely helpful.]*

## Target Audience

This document is intended for use as a reference by the development team, maintenance team to define which tests are to be run and execute from this.

The End User and Operational owner may optionally use this as a reference to understand what black-box testing has been performed as part of entrance criteria to downstream User Acceptance Testing( UAT) in environments where UAT is performed.

## Related documents

These documents contain information related to the information in this document.

| Document Short Name Reference | Document Title | Version (Optional) |
| --- | --- | --- |
| [RelDoc1] | <related-document-name -1> |  |
| [RelDoc2] | <related-document-name -2> |  |
| [ArchDoc1] | <arch-doc -1> |  |
| [ArchDoc2] | <app-onboarding doc> |  |
| [OperationalDoc2] | <Operational run book doc> |  |
| [Test Results] | <Test results> |  |
|  |  |  |

Table 1‑1 Related Documents

## Glossary

| Term/Acronym | Description |
| --- | --- |
|  | *[Description (an example below)]* |
|  |  |
|  |  |
|  |  |
|  |  |

Table 1‑2 Glossary

# Setup and Test environment

This section describes the test environment that the system is to be tested under.

# Tests

### Functional Tests

Functional testing makes sure the system behaves the way users expect. It checks that the main features, like logging in, managing roles, and accessing apps, all work properly. These tests look at how different features interact with each other, not just if they work alone. Functional testing is done through the user interface, just like a real user would use the system. This helps confirm that everything works smoothly together, and that the system is ready for more advanced testing or to be released.

|  |  |  |
| --- | --- | --- |
| **Test ID** | **Purpose** | **Expected Result** |
| FN01 | Verify that App Admins can log in with correct credentials | App Admins are redirected to the dashboard after login |
| FN02 | Verify login fails with incorrect password | Error message is shown: 'Invalid credentials' |
| FN03 | Verify that App Admins can assign a role to a user | Selected user is added to the correct role in the app |
| FN04 | Verify that System Admins can create new App Admins and assign them to an app | New App Admin is created and assigned to a designated app |
| FN05 | Verify that App Admins can create new roles in an app | Role appears in the app’s role list after creation |
| FN06 | Verify that App Admins can remove users from roles | User no longer appears under that role in the app |
| FN07 | Verify that App Admins can add new users to their app | New user appears in user list |
| FN08 | Verify that App Admins can see their assigned apps | Apps are listed correctly in the App Admin dashboard view |
| FN09 | Verify that system admins can see all users | Full user list is accessible by system admin |
| FN10 | Verify that system admins can delete a user | User is removed and no longer accessible |
| FN11 | Verify that App Admins can remove users from the app | User is no longer listed in the app and loses access |
| FN12 | Verify that duplicate users cannot be added to the same app | System prevents duplicate entry and shows an error message |
| FN13 | Verify that App Admins can see all users in their app, sorted | Sort results showing all users |
| FN14 | Verify that role changes take effect immediately | Updated roles are reflected as soon as the change is saved |

**Table 3‑1 Functional Tests**

### Usability Tests

Usability testing checks how easy and intuitive the system is to use. It looks at things like layout consistency, button clarity, error messages, and navigation. The goal is to make sure users can interact with the system smoothly and without confusion.

|  |  |  |
| --- | --- | --- |
| **Test ID** | **Purpose** | **Expected Result** |
| US01 | Verify consistent layout across all pages | All pages have matching headers, footers, and styles |
| US02 | Verify intuitive navigation to App Details view | Users can easily find and access the App Details page |
| US03 | Verify button labels clearly indicate action | Labels like “Add Role” and “Remove User” are clear and accurate |
| US04 | Verify error messages are user-friendly | Errors use simple language and guide the user to fix the issue |
| US05 | Verify form fields have helpful placeholder text | Fields like email and name include helpful hints or examples |
| US06 | Verify color contrast meets accessibility standards | Text and background colors meet accessibility guidelines |
| US07 | Verify tab navigation is clear and usable | Users can switch between tabs (e.g., All Users, Roles) smoothly |
| US08 | Verify feedback is shown after actions (e.g., toast or message) | Actions like “Add User” show a success/failure message |

**Table 3‑2 Usability Tests**

### Business Cycle Tests

Business cycle testing simulates actions that happen over longer periods, like a semester or a year. These tests help confirm that the system handles long-term changes, like user role updates, account archiving, and system resets between cycles.

|  |  |  |
| --- | --- | --- |
| **Test ID** | **Purpose** | **Expected Result** |
| BC01 | Simulate user role changes across two semesters | Role updates persist and reflect correctly over time |
| BC02 | Simulate archiving old user accounts | Archived users are removed from the app interface |
| BC03 | Test semester-end user removal and role reassignment | Previous users are removed, and new users assigned |
| BC04 | Simulate reactivation of inactive user accounts | Reactivated users regain previous roles and access |
| BC05 | Simulate app role structure changes between semesters | Old roles are replaced or updated without conflict |

**Table 3‑5 Business Cycle Tests**

# Non-Functional Tests

### Scale Tests

Scale testing checks if the system can handle a large number of users, roles, or actions. It helps determine whether the system continues to work properly when scaled up to expected real-world usage.

All functional and usability tests are performed under the following scale constraints:

|  |  |  |
| --- | --- | --- |
| **Constraint** | **Constraint Description** | **Size/Amount** |
| ST01 | Max users loaded in a single app instance | 1000 users |
| ST02 | Max roles assigned in a single app | 50 roles |

**Table 4‑1 Scale Constraints**

Cross reference to other tests. Which tests are done under the full-scale conditions.

### Performance Tests

Performance testing measures how fast the system responds under normal and heavy use. It checks that pages load quickly, actions are completed promptly, and the system meets the speed standards expected by users.

Performance tests are all run at maximum scale identified in Section 4.1.1

For any metric-based results, the observed metric is captured so that it is understood how close we are to the failure threshold.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Purpose** | **Expected Result** | **Observed Result** |
| PERF01 | Load dashboard with 1000 users | Page loads within 2 seconds | [To be filled after test] |
| PERF02 | Submit a new role assignment | Action completes within 1 second | [To be filled after test] |

**Table 4‑1 Performance Criteria**

### Security Tests

Security testing focuses on making sure that only authorized users can access the system and perform certain actions. It checks for vulnerabilities like unauthorized access, session management flaws, and common attacks such as XSS or CSRF.

|  |  |  |
| --- | --- | --- |
| **Test ID** | **Purpose** | **Expected Result** |
| SE01 | Verify unauthorized users cannot access admin pages | Unauthorized users are redirected or blocked |
| SE02 | Verify users cannot access app details for apps they don’t belong to | Access is denied with a 403 or redirection |
| SE03 | Verify that user sessions expire after inactivity | User is logged out after a set period |
| SE04 | Verify system prevents XSS attacks in text fields | Inputs are sanitized and scripts do not execute |
| SE05 | Verify that CSRF protection is enabled for form submissions | Unauthorized form submissions are rejected |
| SE06 | Verify that users cannot assign themselves elevated roles | System blocks unauthorized role changes |
| SE07 | Verify login page uses HTTPS | Passwords are sent securely and not in plain text |

**Table 4‑2 Security Tests**

### Compatibility Tests

Compatibility testing ensures the system works correctly across different browsers, devices, and screen sizes. It helps verify that users have a consistent experience, no matter what technology they use to access the app.

|  |  |  |
| --- | --- | --- |
| **Test ID** | **Purpose** | **Expected Result** |
| CO01 | Verify system functions in Google Chrome (latest) | All features work correctly with no layout issues |
| CO02 | Verify system functions in Mozilla Firefox (latest) | All features are usable with no performance issues |
| CO03 | Verify mobile layout on iOS Safari | App layout adapts responsively and remains usable |
| CO04 | Verify system functions in Microsoft Edge | All UI and functionality work the same as Chrome |
| CO05 | Verify behavior on a 1080p vs. 4K screen | Content scales correctly without layout breaking |

**Table 3‑3 Compatibility Tests**

# Conclusion

### Test Results and Analysis

**Total Tests Executed: 40+**

* Passes: ~35
* Fails: 5
* Not Run: 3 (Performance-related)

**Severity Breakdown:**

* 🟥 High: 2 (SE-1, SE-3)
* 🟧 Medium: 2 (US-1, BC-2)
* 🟨 Low: 1 (US-5)

**Overall Quality Assessment:**

The system works well overall and is easy to use in most areas. Most features related to managing apps, users, and roles function as expected. Adding users, assigning roles, and using the form submissions are smooth and do not cause problems. The design of the system makes it simple to understand and interact with for both System Admins and App Admins.

However, there are some problems that affect the usability and security of the system. A few pages look different from the rest, which can confuse users. Some forms are missing helpful placeholder text, which makes them harder to fill out. More importantly, there are serious security issues. For example, App Admins can access areas that should be restricted to System Admins, and sessions do not expire after inactivity, which puts user data at risk.

These issues should be fixed as soon as possible, especially the ones related to security. Fixing them before the system is fully released will improve the quality and make it safer and more reliable for everyone who uses it.

Approvals

This document has been read and approved by the following people, responsible for its implementation. Approval is indicated by an email showing approval. Those approving below indicate that the contents of this document are correct and complete and agree to their implementation:

| Title | Name | Approval |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

History

| Version | Status | Date | Author | Reason for changes |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |