TLC

System Test Plan

Document Category: TP

Version:<1.0>

Status:<>

Last Updated:<07/03/2020>

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 8](#_Toc1725866)

[3.1.3 Business Cycle Tests 9](#_Toc1725867)

[4 Non-Functional Tests 10](#_Toc1725868)

[4.1.1 Scale Tests 10](#_Toc1725869)

[4.1.2 Performance Tests 10](#_Toc1725870)

[4.1.3 Security Tests 11](#_Toc1725871)

[4.1.4 Compatibility Tests 12](#_Toc1725872)

[Approvals 14](#_Toc1725873)

[History 15](#_Toc1725874)

# Introduction

## Purpose

The purpose of this document is to describe the system test plan for TLC 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 |
| --- | --- |
| Session | A student in the queue |
| Active Session | A student being helped |
| Queue | Line in which students wait |
|  |  |
|  |  |

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 is a quality assurance process and a type of black-box testing.

|  |  |  |
| --- | --- | --- |
| **Test ID** | **Purpose** | **Expected Result** |
| FN01 | Teacher can see their queue. | Teacher can see the queue for their department. Can’t see queues for other departments or other teachers. |
| FN02 | Student can see active queues. | Student can see a summary for all active queues. Can see how many are waiting and the scheduled time for the queue. |
| FN03 | Student can be added to queue. | When teachers add a student, they are added into their queue. Students can see more details if they were to login to their own account. |
| FN04 | Student session status can change | Student sessions can be active, waiting, canceled or completed. These changes will be made by the teacher when moving students, finishing a session or removing them. Students can add themselves or remove themselves. |
| FN05 | Student can be removed | Student can remove themselves, or be removed by teacher from the queue |
| FN06 | Teacher can edit where the student is in queue | Position and status can be changed by teacher when dragging sessions. |
| FN07 | Users can login | Login gives appropriate permissions. Meaning student has access to their dashboard, and schedule. Teachers will have access to their queue, student mode and schedule. |
| FN08 | Users can logout | Users are logged out, can’t be logged back in on refresh or when back button is pressed. |
| FN09 | Users are given feedback on invalid login | User is not logged in, given feedback |
| FN10 | Users can see schedule | Users can see the schedule to know who and what subject will be available at what time |
| FN11 | Teacher can edit the schedule | Teachers can add themselves or remove their subject in the schedule |
| FN12 | Teacher can go into student mode for students to add themselves to queue in the TLC | Students added. Can be seen from teacher queue, and from the students dashboard. |

**Table 3‑1 Functional Tests**

### Usability Tests

It is difficult for those working on a system to see the flaws in it. Therefor we use usability testing in an attempt to evaluate a product’s usability. It is a test in which users test the product and their behaviour and feedback is noted based on user interface, navigation and ease of use.

|  |  |  |
| --- | --- | --- |
| **Test ID** | **Purpose** | **Expected Result** |
| US01 | Student adds themselves in UQ mode | Student line up, add themselves for the subject they’d like help with, then added to the queue.  Students shouldn’t have issues adding themselves. |
| US02 | Teachers manage their queue | Teachers can:   * Add student * Remove student * Move student   Should be easy and intuitive. User interface should be obvious that queue is draggable. |
| US03 | Student can see their position in the queues they’ve signed up for | After logging in, students can see a list of queues they are in and can remove themselves. This tests ease of use for students to see their information in a queue or multiple queues. |
| US04 | Users can see the schedule | Users can see the schedule and the different subjects. User interface of the schedule will be tested for feedback. |
| US05 | Teachers can edit the schedule | Teachers can edit the schedule, meaning add or remove time slots for their subject. This will test ease of use for when it comes to teachers editing schedule. |
| US06 | Users can logout | User can logout. Quick and easy, meaning users shouldn’t look for a long time for where to logout. Ease of use testing. |

**Table 3‑2 Usability Tests**

### Business Cycle Tests

Business cycle tests are meant to test activities or processes performed on the system over time.

|  |  |  |
| --- | --- | --- |
| **Test ID** | **Purpose** | **Expected Result** |
| BC01 | Teacher manages queue | Teacher can add, move and remove or complete student sessions. Make sure that multiple combinations of actions can be performed simultaneously by users. |
| BC02 | Student adding themselves | Students line up, add themselves to a queue that is currently active and add themselves once. Students can add themselves to multiple queues. |
| BC03 | Data report download | Teachers will choose a time frame and data, or a summary of the data can be downloaded. |

**Table 3‑5 Business Cycle Tests**

# Non-Functional Tests

### Scale Tests

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

|  |  |  |
| --- | --- | --- |
| **Constraint** | **Constraint Description** | **Size/Amount** |
| 1 | Max users at a time on the service. | Around 20 users should be more than enough. |
| 2 | Requests per second when trying to download or interact with reports. | 3 teachers, no less that 2 seconds for response |
| 3 | Acceptable response time, web server requests and responses incorporated (React) | Should not take more than 2 seconds for an action to be performed. |
| 4 | Requests per second when trying to perform actions that use React | 20 users |

**Table 4‑1 Scale Constraints**

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

### Performance Tests

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.

The performance metrics are:

* Response time for React processes
* Response time when a lot of React processes are running simultaneously
* Response time for data reports or summary reports
* Response time when a lot of data reports or summary reports are running simultaneously
* Number of users online at a time

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Purpose** | **Expected Result** | **Observed Result** |
| 1 | Load and scalability testing the number of users online. Test the maximum number of users online we can have at a time. Then we would use endurance testing for that maximum amount. | No bugs or decrease in performance. |  |
| 2 | Load, scalability and endurance testing when multiple users make an action using React | No bugs or decrease in performance. |  |
| 3 | Perform stress testing on the system in general. | Find hardware that fails in order. |  |
| 4 | Volume, endurance, spike and load testing the amount of data that can be reported as well as the number of requests we can have for reports simultaneously. | No bugs or decrease in performance. |  |

**Table 4‑1 Performance Criteria**

### Security Tests

Test for vulnerabilities and make sure the system is secure, as well as make sure users have appropriate access to content.

|  |  |  |
| --- | --- | --- |
| **Test ID** | **Purpose** | **Expected Result** |
| SE01 | Login must be authenticated, as well as no default or test login accounts available during production. | Not logged in if invalid or unauthenticated or default/test login account, otherwise users are given access to appropriate content. |
| SE02 | Users can’t press back or refresh page to login after a user has logged out. | Refresh or back does not give access to system if a user has logged out. |
| SE03 | SQL can’t be injected into any forms. | Data is not sent back when an attempt at SQL injection is made. |
| SE04 | No information is given in the URL. | No information such as student ID or anything else should be in the URL, since it might leak information and make session high jacking a possibility. |
| SE05 | Make sure Cross Site Scripting is not possible. | JavaScript tags can’t be injected into forms. HTML shouldn’t be entered as well. |

**Table 4‑2 Security Tests**

### Compatibility Tests

Compatibility testing involves testing the hardware, operating system, software, network, browser, devices, mobile as well as different versions to make sure they work well together.

|  |  |  |
| --- | --- | --- |
| **Test ID** | **Purpose** | **Expected Result** |
| CO01 | Test if all browsers work well with React, including Internet Explorer. | System works perfectly on all browsers. |
| CO02 | Test if the system will work in a development or production environment. | System works perfectly no matter the environment. |
| CO03 | Test if system works on different hardware. | In the case of a change in hardware, such as upgrading the server, system should work nonetheless. |

**Table 3‑3 Compatibility Tests**

# Conclusion

### Test Results and Analysis

Your conclusion after the completion of the test execution(s).

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 |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |