CSP – Regression

Automation

Test Plan

Prepare By

**Narayana Kaluri**

Prepared On

**December 09, 2019**

**Table of Contents**

[1. Change history 4](#_Toc26900196)

[2. Introduction 4](#_Toc26900197)

[2.1 Automation Testing DTP Overview 4](#_Toc26900198)

[3 Test Description 5](#_Toc26900199)

[3.1 Test Identification 5](#_Toc26900200)

[3.2 Test Purpose and Objectives 5](#_Toc26900201)

[3.3 Entry Criteria 5](#_Toc26900202)

[3.4 Exit Criteria 6](#_Toc26900203)

[3.5 Suspension criteria 6](#_Toc26900204)

[3.6 Resumption criteria 6](#_Toc26900205)

[3.7 Pass/Fail Criteria 6](#_Toc26900206)

[4 Test Scope 7](#_Toc26900207)

[4.1 Functions to be tested 7](#_Toc26900208)

[4.2 Functions not to be tested 7](#_Toc26900209)

[5 Test Approach 7](#_Toc26900210)

[5.1 Analysing feasibility study 8](#_Toc26900211)

[Based on the manual test cases provided by team, a feasibility study will be performed and cases will be segregated into module wise and repetitive steps will be converted into reusable methods. 8](#_Toc26900212)

[5.2 Framework 8](#_Toc26900213)

[A Hybrid framework will be designed which makes the scripting and maintenance very ease. Using following techniques. 8](#_Toc26900214)

[5.2.1 Page Object Pattern 8](#_Toc26900215)

[5.2.2 Modular Functions 8](#_Toc26900216)

[5.2.3 Common Functions 8](#_Toc26900217)

[5.2.4 Reports 8](#_Toc26900218)

[5.3 Design test data 9](#_Toc26900219)

[5.4 Scripting 9](#_Toc26900220)

[Scripting will be completed as mentioned in the schedule for each test case with standards and naming conventions mentioned in Test Script template. 9](#_Toc26900221)

[5.5 Review 9](#_Toc26900222)

[5.6 Execution 10](#_Toc26900223)

[5.7 Source Version Control 10](#_Toc26900224)

[5.8 Requirement Traceability Matrix 10](#_Toc26900225)

[5.9 Maintenance 10](#_Toc26900226)

[5.10 CI/CD Automation deployment approach 10](#_Toc26900227)

[5.11 Test deliverables 11](#_Toc26900228)

[6 Test Preparation Specifications 11](#_Toc26900229)

[6.1 Test Environment 11](#_Toc26900230)

[6.2 Test Team Roles and Responsibilities 11](#_Toc26900231)

[6.3 Test Team Training Requirements 11](#_Toc26900232)

[7 Estimation 12](#_Toc26900233)

[8. Test Issues and Risks 12](#_Toc26900234)

[8.1 Issues 12](#_Toc26900235)

[8.2 Risks 12](#_Toc26900236)

# Change history

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Reference** | **Version** | **Author** | **Date** | **Change Description** | **Approved By** |
|  | 0.1 | Narayana Kaluri | 09/12/2019 | Initial Draft | Sohan Belgur |
|  | 0.2 | Rajiv Sai Ballapuram | 3/4/2020 | Added system requirements |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# Introduction

|  |  |
| --- | --- |
| Project name: | CSP |
| Project-ID: | To be updated |
| Customer: |  |
| Product owner: |  |
| Project Manager: |  |
| Testers: | Sohan Belgur, Likhitha Joshi, Rajiv Sai Ballapuram, Madhuri |
| Project Architect : |  |

## Automation Testing DTP Overview

Automation Testing Detail Test Plan (ADTP) will identify the specific tests that are to be performed to ensure the quality of the delivered product. It will cover information for automated testing during the Regression/Enhancement Phase of the project and will map to the specification or requirements for the project provided my manual team test cases. This mapping is done in conjunction with the Traceability Matrix document that should be completed along with the ADTP and is referenced in this document.

The objectives of ADTP are:

• Identify automatable test cases.

• Describe the test to be executed.

• Identify and assign a unique name/number for each specific test script.

• Describe the scope of the testing.

• List what is and is not to be tested.

• Describe the test approach detailing methods, techniques, and tools.

• Identify issues and risks.

# Test Description

## **3.1 Test Identification**

ADTP is intended to provide information for automation testing for the Client Service Portal Application. The test effort may be referred to by its PROJECT REQUEST (PR) number and its project title for tracking and monitoring of the testing progress.

## 3.2 Test Purpose and Objectives

The primary objective of automation testing is to: assure that the system meets the full requirements, including quality requirements (functional and non-functional requirements) and fit metrics for each quality requirement and satisfies the use case scenarios and maintain the quality of the product during regression/Enhancement phase of the project.

## Entry Criteria

* The environment for testing, including databases, application programs, and connectivity has been defined, constructed, and verified.
* All the necessary documentation, design, and requirements information should be available that will allow automation testers to complete the scripting on time. Manual test cases along with required test data and pre-conditions are ready.
* All the standard software tools including the testing tools must have been successfully installed and functioning properly.
* Proper test data is available/ can be created using test approach techniques provided in the document.
* QA resource/ POC with sound knowledge of functionality
* Reviewed test scenarios, test cases and RTM
* All required software are installed for scripting.

## Exit Criteria

In establishing the exit/acceptance criteria, all automated test cases have been executed as documented and automation report with stable scripts. The percent of successfully executed test cases met the defined criteria. Recommended criteria: No Critical or High severity problem logs remain open and all medium problem logs have agreed upon action plans; successful execution of the application to validate accuracy of data, interfaces, and connectivity.

## Suspension criteria

* The build contains any critical defects which seriously or limit testing progress.
* Significant change in requirements suggested by client
* Software/Hardware problems
* Assigned resources are not available when needed by test team.

## Resumption criteria

Resumption will only occur when the problem(s) that caused the caused the suspension have been resolved

## Pass/Fail Criteria

The results for each test must be compared to the pre-defined expected test results, as documented in the ADTP (and manual test case). The actual results are logged in the Test Report detail within the Detail Test Plan if those results differ from the expected results test case is marked as failed. If the actual results match the expected results, the Test Case can be marked as a passed item, without logging the duplicated results.

A test case fails if the actual results produced by its execution do not match the expected results. The source of failure may be the application under test, the test case, the expected results, or the data in the test environment. Test case failures must be logged regardless of the source of the failure.

Any bugs or problems will be logged in the DEFECT TRACKING TOOL with the help of manual team.

The responsible application resource corrects the problem and tests the repair. Once this is complete, the tester who generated the problem log is notified, and the item is re-tested. If the retest is successful, the status is updated and the problem log is closed.

If the retest is unsuccessful, or if another problem has been identified, the problem log status is updated and the problem description is updated with the new findings. It is then returned to the responsible application personnel for correction and test.

# 4 Test Scope

The scope of testing identifies the items which will be tested and the items which will not be tested within the Automation Phase of testing.

## 4.1 **Functions to be tested**

* GUI
* Search and Filters Logic
* API (If Applicable)

## 4.2 **Functions not to be tested**

* Security
* Performance
* Functional

# 5 Test Approach

The mission of Automated Testing is the process of identifying recordable test cases through all appropriate paths of a website, creating repeatable scripts, interpreting test results, and reporting to project management. The automation test team will focus on positive testing and will complement the manual testing undergone on the system. Automated test results will be generated, formatted into reports and provided on a consistent basis to project management.

Content testing focuses on whether the content of the pages match what is supposed to be there, whether key phrases exist continually in changeable pages, and whether the pages maintain quality content from version to version.

Accuracy and consistency testing focuses on whether today’s copies of the pages download the same as yesterdays, and whether the data presented to the user is accurate enough.

Completion of automated test cases is denoted in the test cases with indication of pass/fail and follow-up action.

## **Analysing feasibility study**

Based on the manual test cases provided by team, a feasibility study will be performed and cases will be segregated into module wise and repetitive steps will be converted into reusable methods.

## **Framework**

A Hybrid framework will be designed which makes the scripting and maintenance very ease. Using following techniques.

### Page Object Pattern

Every window/page class will be created which include all the controls available in them which can be reused anywhere in the project

### Modular Functions

Using POM every class will contain modular functions which perform actions in the page using various parameters, these can be reused while scripting.

### Common Functions

Methods reused across the projects will be in common function which include reading data from Excel, Querying DB, Handling windows etc..

### Reports

Each test script will record steps in logger using Extent Report along with screenshot and consolidated report will be generated in specified path.

* + 1. **Separate scripts**

Test scripts will be maintained separately for each module in individual folders as per their priority.

* + 1. **Nunit Testing framework**
* Tests can be run from a console runner, within Visual Studio through a Test Adapter or through 3rd party runners.
* Tests can be run in parallel.
* Strong support for data driven tests.
* Assertions can be performed.

## **Design test data**

Test data will be generated by respective QA on test database based on scenarios and Test cases using SQL procedures or through API calls which will be read in scripts by querying.

## **Scripting**

Scripting will be completed as mentioned in the schedule for each test case with standards and naming conventions mentioned in Test Script template.

## **Review**

* Peer review will be conducted for test scripts and test matrix by QA Lead
* Any comments or suggestions on test scripts and test coverage will be provided by reviewer respective Author of Test Case and Test Matrix
* Suggestions or improvements will be re-worked by author and will be send for approval
* Re-worked improvements will be reviewed and approved by reviewer

## **Execution**

* Test cases will be executed by respective QA on client's development/test site based on designed scenarios, test cases and Test data.
* Test result (Actual Result, Pass/Fail) will provided in test report
* QA will be logging the defect/bugs in Word document, found during execution of test cases. After this, QA will inform respective team about the defect/bugs.

## **Source Version Control**

QA Team will be checking in code/scripts into TFS on regular intervals after code review and execution along with reports.

## **Requirement Traceability Matrix**

QA will be preparing for each function to undergo testing by automation, the Test Case is identified. Automated Test Cases are given unique identifiers to enable cross-referencing between related test documentation, and to facilitate tracking and monitoring the test progress.

As much information as is available is entered into the Traceability Matrix in order to complete the scope of automation during the regression/Enhancement Phase of the test.

## **Maintenance**

Regular executions will be performed on dedicated machine to ensure the stability of scripts and will be maintained in case of any discrepancies.

## **CI/CD** Automation deployment approach

Building automation script is 70% of your total automation effort but scaling automation infrastructure, practicing and implementing successful DevOps is also challenging.

Docker/Private agent + Kubernetes/Agent pool (TFS) + Jenkins + DevOps best practices, will be used as DevOps pipeline.

## **Test deliverables**

* Test Scripts and their maintenance.
* Test reports.
* CI/CD configuration along with batch files.

# Test Preparation Specifications

## Test Environment

**Hardware**:

* 16 GB RAM,
* 256 GB ROM,

**Software**:

* OS: Windows 10 with admin access ,Developer Mode Enabled,
* Inspect tool
* Visual Studio enterprise
* Appium
* WinAPPDriver tool
* Selenium WebDriver
* IEServer
* Nunit
* Extent Reports
* Docker
* Jenkins
* TFS build agent access
* CSP installed

## Test Team Roles and Responsibilities

|  |  |  |
| --- | --- | --- |
| Role | Name | Responsiblities |
| QA Manager | VenkataKrishna Hariharan | To be Updated |
| QA Lead | Sohan Belgur | To be Updated |
| QA | Rajiv Sai Ballapuram | Framework Development  Scripting and Reviewing |
| QA | Likhitha Joshi | Scripting |
| QA | Madhuri | Scripting |

## Test Team Training Requirements

* + - Windows Application Driver tool training to the QA automation team.

# Estimation

To be Updated (Based on Manual test cases)

# Test Issues and Risks

## Issues

The table below lists known project testing issues to date. Upon sign-off of the Detail Test Plan, this table will be maintained, and these issues and all new issues will be tracked.

To be Updated (Based on Manual test cases)

## Risks

To be Updated (Based on Manual test cases)