**Test Strategy and Plan**

# Introduction

This Test Plan document describes the process for conducting testing on the Assignment assigned on Customer Onboarding Application

The customer Onboarding Application enable customization and a tenant who buys this application can customize the fields to be captured, the validations to be performed, number of views, sub Views, and the workflow for approval. The entire configuration is stored as json in Mongo. When a user logs in based on this configuration dynamic views are presented so that the Customers can be onboarded.

# Objective

This document describes the plan for testing the Customer onboarding Application.

* List the recommended test requirements
* Recommend and describe the testing strategies to be employed
* Identify the required resources
* List the deliverable elements of the test activities

# Scope

## In Scope

### Functional Requirements

The following are the list of modules available in the system. The scope is subject to change based on the final requirements and the testability aspect.

Creating user Functionality: user should be created when valid inputs are given

Ex: Name, Address, View id, Status, workflow id, Onboard

Get user details: Getting the user information based on the given id {id}

Validations to be performed on number of views, sub Views, and the workflow for approval

Root for the API Is placed in Repository (Ex: GitHub)

Data is stored in JSON (java script object notation) Structure

## Out of Scope

Following aspects of testing are considered as functional and out of scope:

* Usage of automation tool for regression testing
* Non functional requirement testing and UI based testing

Following aspects of testing are considered as Non-functional and out of scope:

* Volume testing - This effort is to make sure that System performance is tested / verified, to meet/confirm industry standards. Not in scope due to dependency on Hardware.
* Advanced Security testing
* Usage of Security testing tool
* Advanced Performance testing

Any requirement/topic which is **NOT** specified in Functional requirements document is deemed to be out of scope.

# Assumptions & Dependencies

## Assumptions

* Test environment must be ready before the beginning of the test execution.
* Dedicated QA resource is assigned to perform the testing operations.
* QA resource is educated with the complete flow of the application or the area under test.
* The estimations and the deadline are defined explicitly.
* The necessary test case documents are prepared and reviewed.
* The requirements are freezed and all the clarifications and impacts are analyzed.
* The compatible system is in place.

## Dependencies

* The source code must be unit tested and provided within the Project schedule
* Providing Requirements and other necessary documentation in English
* Providing timely Review comments / Feedbacks on deliverables in order to avoid any delays in the project schedule

# Test Requirements

Following types of testing will be performed when a build is ready to be tested:

* Smoke testing
* Usability testing
* System/Integration
* API testing
* Compatibility testing

# Test Approach

## Smoke Testing:

It is a non-exhaustive testing which is performed when a new code is given. Smoke testing is done to touch every part of the application without going deep into functionality

|  |  |
| --- | --- |
| Test Objective: | 1. The smoke test will be performed based upon the Unit test results provided. The smoke test cases will be executed and all the basic functionalities of the application will be verified for confirming the stability of the build |
| Entry Criteria | 1. QA Release process to be followed 2. A minimum of 90% of Unit test cases should pass [First Pass Yield - FPE] 3. build is installed and the test environment is set |
| Technique: | 1. A set of important test cases from system test cases have been identified as Smoke test cases 2. Execute the smoke test cases 3. Check that all the basic functionalities of the application are working and the build is ready for further testing |
| Exit Criteria: | 1. All identified Smoke test cases should pass |

## Integration/System testing

Integration testing is a method in which individual software modules are combined and tested as a group

## Regression Testing:

Regression testing is re-execution of test cases from existing test suites to ensure that software changes have no unintended side-effects. Regression testing is initiated when bug fixes are done for a recognized problem. It is a quality control measure to ensure that the newly modified code still complies with its specified requirements and that unmodified code has not been affected by the maintenance activity.

|  |  |
| --- | --- |
| Test Objective: | 1. Regression test is carried to make sure the bug fix has not affected any other part of the application |
| Entry Criteria | 1. The smoke and system testing has passed 2. The bug fixes are done and new build installed on the test server |
| Technique: | 1. Execute all the functional test cases 2. To check whether the code fixes have not changed the functionality of other parts of the application |
| Exit Criteria: | 1. All the test cases identified for regression testing should pass and all bugs identified have been reported |

## Compatibility testing:

Compatibility testing is conducted on the application to evaluate the application's compatibility with the computing environment such as Operating systems, web browsers a. By performing compatibility testing it is ensured that the application behaves and responds the same way across different computing environment.

## Usability Testing:

Usability testing is a method in which testing the ease with which users can learn and use a product (But Not Applicable to the given assignment)

|  |  |
| --- | --- |
| Test Objective: | 1. The look and feel of the application should be user friendly |
| Entry Criteria | 1. Unit test cases should be passed 2. Release process to be followed |
| Technique: | 1. Every page or screen in the application should have the CSS implemented 2. Alignment and Fonts should be according to GUI standards |
| Exit Criteria: | 1. All the usability test cases have been executed. |

# Test Execution

* complete test execution is dependent on testcase readiness, environment factors, data availability, and time available, risk based testing
* 80 % completion of test case execution to consider that a build is tested

# Deliverables

Below listed are the documents from the Testing team during the entire Testing effort:

|  |  |  |
| --- | --- | --- |
| **Deliverable** | **Owner** | **Review / Distribution** |
| Test Plan & Strategy | QA/Team lean | Zycus |
| System Test Cases/Scripts | Test Team | Zycus |
| Test summary report or code coverage report | Test Team | Zycus |

# Resources

## Human Resources

|  |  |  |
| --- | --- | --- |
| **Human Resources** | | |
| **Human Resources** | **Minimum Resources Recommended** | **Specific Responsibilities/Comments** |
| Narendra | 2 | Test execution System test case) and Defect management, Functional Testing, Manual testing, Smoke Testing |

## Software

|  |  |  |
| --- | --- | --- |
|  | **Client** | **Server** |
| OS | Windows 7 64 bit |  |
| Browser | Chrome |  |
| API | JSON | Flask |
| Tool | Request, jsonify |  |
| Language | Python |  |

## Hardwar

|  |  |
| --- | --- |
| **Client** | **Server** |
| Intel core i5 processor 6 GB RAM |  |
|  |  |

# Test Schedule and Milestones

Schedule and Milestone information will be as specified in Zycus Team:

# Risk Management

The test plan and test schedule are based on the current Requirements document.

# Defect Management

* Defect reports are generated on daily basis during execution and they are shared with the stakeholders
* Defect Triage meetings will be supported by relevant details/steps/screen shots/other necessary items from the Testing team

Below mentioned steps indicate the Defect flow in Bug tracker:

* When a new bug is created then the status will be ‘Open’
* The tester can assign a bug to the developer
* After assigning the bug to a developer then the developer will change the status of the bug to ‘Fixed’
* If the bug cannot be simulated and if there are any clarifications awaiting on the bug then the developer will change the status to ‘Not A Bug’
* If the bug cannot be fixed in current build due to delay in clarification from customer or don’t have enough time then developer will change the status to ‘On hold’
* When the Fixed bug is retested by testing team then the status will be changed to ‘‘verify’ if it is working else it will be ‘Re-Open’.
* After 3 days from the date of verified bugs, system will change the status to ‘Closed’.

### Defect Severity and priority:

Severity is the indication of the impact of the fault on Test and/or in Production.

|  |  |  |
| --- | --- | --- |
| **Defect Severity** | **Guidelines** | **Priority** |
| 1 – Critical | All/most Test activities suspended due to fault. System un-usable and/or major user groups prevented for using system. No workaround available. | 1 |
| 2 – High | Test Activities for module suspended. Major module of system un-usable and/or groups of users unable to work. Workarounds may be available. | 2 |
| 3 – Medium | Test script suspended but other testing can continue. Individual functions affected and/or individual users prevented from completing tasks. Workarounds available. | 3 |
| 4 – Low | Minor fault or Cosmetic Error that has no impact on testing schedule. Users can continue working. | 4 |

# Tools

|  |  |  |
| --- | --- | --- |
| **Areas in which Tools will be used** | **Tool** | **Vendor/In-house** |
| Defect Tracking | Not Applicable | In - house |
| Repository | GitHub | In - house |
|  |  |  |

# Release process

## release to Zycus Team

provide the following deliverables as part of Release process:

* Build – Version and Path
* Release notes - included requirements, Open issues, any mandatory information for Test team
* System test case
* API test case
* Test plan and Strategy document
* Code coverage report
* Scripts shared to team on Github repositary

## Release to Client/Internal Customer

* Test execution results
* Release notes/Certification mail

# Entry/Exit Criteria

## Entry

To begin with testing effort for any build/release, following conditions are to be met

* Release process followed
* Smoke test results – 80% Pass

## Exit

To stop testing effort for any build/release deemed to be final/deliverable to Client

* No Open High/Critical defects
* All the listed testing types are executed completely

# Document References

* SRS: - QA\_Assignment\_testing (shared in Github)