LIFERAY FORMS

Test Plan <V1.0>

Document Change History

Version Number	Date	Contributor	Description
V1.0	08/29/2021	Lucas Florêncio de Brito	First version

Table of Contents

1 Introduction

This Test Plan has been created to communicate the test approach to team members. It includes the objectives, scope, schedule, risks and approach. This document will clearly identify what the test deliverables will be and what is deemed in and out of scope.

1.1Objectives

The purpose of this document is to define:

- The test scope, focus areas and objectives
- The test responsibilities
- The test strategy for the levels and types of test for this release
- The entry and exit criteria
- The basis of the test estimates
- Any risks, issues, assumptions and test dependencies
- The test schedule and major milestones

1.2Team Members

Name	Role
Lucas Florêncio de Brito	Test Engineer

2 Scope

This test plan consists of tests in the system level. It is assumed that the unit test has already provided, extensive coverage of the source code and testing of all module interfaces. It is also assumed that the integration tests have already been done.

The following strategies will be approached:

- Functionality testing: Functional testing is aimed to ensure that each function of the form operates in conformance with the requirement specification;
- Usability testing: helps to define user ability to learn to operate, prepare inputs for, and interpret outputs;
- User Interface testing: is provided to verify the graphic user interface meets the specifications;
- Compatibility Testing: testing is performed to test each one of the supported software and hardware configurations;
- Cross-platform testing allows evaluating the work in different operational system and browsers;
- Performance testing: is aimed to determine how a system performs in terms of responsiveness and stability under a certain load. Performance website testing methods contain stress testing, load testing, stability testing, volume testing, page load speed testing.

- Security Testing: is performed to verify the information system protects data and maintains functionality as intended.

The tests will be performed in multiple browsers (Microsoft Edge, Internet Explorer, Firefox, Chrome, Opera, Safari) and using Windows, MacOS, and Linux.

3 Assumptions / Risks

3.1 Assumptions

This section lists assumptions that are made specific to this project.

1. All the Test Engineers have computer and devices needed to perform the test.
2. Test environment is set.
3. Software and frameworks used to write test cases, bug tracking, report bugs and test automation are available to all team member.

3.2 Risks

The following risks have been identified and the appropriate action identified to mitigate their impact on the project. The impact (or severity) of the risk is based on how the project would be affected if the risk was triggered. The trigger is what milestone or event would cause the risk to become an issue to be dealt with.

#	Risk	Impact	Trigger	Mitigation Plan
1	Changes made to the functionality may negate the tests already written and we may lose test cases already written.	High	Test cases need to be rewritten.	Each iteration, functionality will be closely monitored. Priorities will be set and discussed by stakeholders.
2	Team members affected by Covid-19	Medium	Less productive	The team member will work from home during this project.
3	Insufficient resources	Low	Less productive	All the resources will be acquired before the project start

4 Test Approach

The project is using an agile approach Scrum, with weekly iterations. At the end of each week the requirements identified for that iteration will be delivered to the team and will be tested.

Manual, automated and Exploratory testing will be executed during the plan. Tests for planned functionality will be created and added to as we get iterations of the product.

4.1Test Automation

Automated unit tests are part of the development process, automated performance test and functionality test will be used too. We will use Selenium WebDriver with Java to conduct the functional tests and Jmeter to do the performance test.

5 Work Plan

The parties are agreed to follow the next work plan:

1. Test plan preparation
2. Test plan approval
3. Functional testing and bugs reporting
4. Performance testing and bugs reporting.
5. Daily reports preparation
6. Final report preparation

6 Test Procedure

Various aspects of the tested software should be checked; this requires executing of different testing types.

The main testing type that would be executed:

- Functional Testing
- UI Testing
- Usability Testing
- Compatibility Testing (using different web browsers)
- Regression testing
- Performance Testing

7 Bug Reporting

Bug reports are created in order to provide the development team and the project managers with exhaustive information about the discovered defects. They must be helpful in determining causes of the errors and correcting them.

7.1 Defect Severity

Defect Severity can be classified into four categories:

- Critical (blocker) defects are the failure of the complete software system or of acritical subsystem, and no work or testing can be carried out after the occurrence of the defect. It also applies to data loss failures and with processes that leave inconsistent data stored on the database.
- Major defects (and crashes) are those which also causes failure of entire or part of the system, but there are some processing alternatives which allows further operation of the system. It also applies to the system crashing, or aborting, during normal operation of a non-critical flow.
- Minor defects do not result in failure but causes the system to show incorrect, incomplete, or inconsistent results.
- Trivial defects are small errors that do not affect the functionality: typos, grammar mistakes, wrong terminology, etc.

7.2 The information that is indicated in each bug report

Each bug reported need to have the following information:

- The software product name;
- Version number of the software product;
- Browser on which the test was performed;
- Operational system which the test was performed.

Each report provides the following information about the defect:

- Summary, which is short description of the problem;
- Steps to reproduce the error;
- Expected results;
- Actual results;
- Frequency of the defect occurrence;
- Severity of the defect;
- Additional information about the defect in form of attached screenshots or video records.

8 Resources

The following tools will be used for this project:

Objective	**Name**
Defect Tracking	Jira
Write Test Cases	Dalek

The team members will need computers with different Operational Systems (Windows, Linux, MacOS)