

Liferay Form QA Testing Interview

Test Plan

CHANGELOG

Version	Change Date	Prepared by	Changes description
V01	15/11/2021	Pablo Mantas	First version

Table of content

1. INTRODUCTION	3
1.1. SCOPE	3
1.1.1. In Scope	3
1.1.2. Out of Scope	3
1.2. QUALITY OBJECTIVE.....	3
2. TEST STRATEGY.....	4
2.1. Test Objectives.....	4
2.2. Test Assumptions	4
2.3. Test Principles	4
2.4. Data Approach.....	5
2.5. Levels of Testing	5
2.5.1. Exploratory	5
2.5.2. Functional Test	5
2.5.3. Non-functional test	5
2.6. Bug Triage.....	6
2.7. Test Completeness	6
2.8. Test Deliverables	7
3. EXECUTION STRATEGY	8
3.1. Test Cycles	8
3.2. Validation and Bug Management.....	8
4. TEST MANAGEMENT PROCESS.....	9
4.1. Test Management Tool	9
4.2. Test Design Process.....	9
4.3. Test Execution Process	9
4.4. Test Risks and Mitigation Factors.....	9
5. RESOURCE AND ENVIRONMENT	11
5.1. Testing Tools.....	11
5.2. Test Environment	11

1. INTRODUCTION

This test plan describes the testing approach and overall framework that will drive the testing of the Liferay Form. It is a release candidate deployed in the test environment. The document introduces:

- Test Strategy:
- Execution Strategy:
- Test Management:

1.1. SCOPE

1.1.1. In Scope

- All the tests to be carried out are black box, focuses on input and output.
- The correct visualization and usability of the form will be tested in different browsers and resolutions.
- The texts for different regions will be checked (US & PT).
- The data entry will be validated based on the title/description of the field to be filled.
- The response of the form will be validated before filled/unfilled mandatory fields.

1.1.2. Out of Scope

- Unit tests are not in the scope of the test plan.
- Verification of the sending of the data entered by the user.
- Performance tests such as scalability will not be done.
- Security tests will not be carried out, although risks will arise if the minimum standards of web development are not met.

1.2. QUALITY OBJETIVE

The objective is identified bugs/issues before to go to production environment and establish an execution strategy that allows agile development with the highest product quality. The coverage of this plan is 100% for the specified scope between manual and automatic tests.

2. TEST STRATEGY

Testing life cycle will include five phases:

- Impact Assessment: Gather inputs from stake holders as feedback for next deployment cycle
- Testing Planning: Testing team come together to update test plan, meeting frequency and testing process.
- Daily meetings: Everyday morning meeting to catch up on the status of testing and report the status of team task in progress.
- Weekly Team Iteration: Review meeting with stake holders to assess the progress and status of US against milestones.
- QA Complete/ Release Readiness: Review the features developed and assess if its are ready to release or not.

2.1. Test Objectives

Will follow Agile testing as methodology to support continuous testing and improve product quality. The objective of this strategy is to detect errors in the behavior of the Liferay form that can be reported to the development team and check their resolution efficiently.

2.2. Test Assumptions

- Exploratory Testing would be carried manually.
- Performance testing is not considered for this estimation.
- All bugs/issues would come along with a snapshot JPEG format.
- The Test Team will be provided with access to Test environment via URL allocated in a corporate site.
- Test environment and preparation activities will be owned by Test Team.
- Test team will manage the testing effort with close coordination with Project PM/BUSINESS ANALYST
- There is no environment downtime during test due to outages or defect fixes.

2.3. Test Principles

- Testing will be focused on meeting the product objectives, cost efficiency and quality.
- There will be common, consistent procedures for all teams supporting testing activities.
- Testing processes will be well defined, yet flexible, with the ability to change as needed.
- Testing activities will build upon previous stages to avoid redundancy or duplication of effort.
- Testing environment and data will emulate a production environment as much as possible.
- Testing will be a repeatable, quantifiable and measurable activity.

2.4. Data Approach

- To carry out the functional tests of the behavior of the form, we will base our data on a user configuration file with different input parameters (Name, Birth date and Text Area)
- The region change tests will be based on a configuration file with texts in each language.
- For non-functional usability tests, we will collect the most recent user statistics based on the six most used browsers and the two most common resolutions per device (Desktop monitor, Laptop, Tablet and mobile).

2.5. Levels of Testing

2.5.1. Exploratory

Exploratory testing is carried out in the application without any test scripts and documentation with the aim of continuously improving the design and execution of tests.

The purpose is detecting critical bugs.

2.5.2. Functional Test

Functional testing will be performed by Testing Team to check the functions of application. The functional testing is carried out by feeding the input and validates the output from the application.

The purpose is to validate whether or not the software is fixed and ready for its release.

Scope: The below excel sheet details at high level the scope of Functional Test

Method: The tests will be divided between manual tests and automated tests with Selenium/Selenium WebDriver. The execution of the automated tests will be carried out periodically, depending on their category, by a Jenkins job.

Test Acceptance Criteria:

- The form must be accessible via the URL provided
- All texts must appear in the language of the region we select.
- You should not submit the form if any of the required fields are empty.
- Inform the user with a descriptive error message in case of an error in the form after submit it.
- No special characters should appear in the name field.
- The date of birth cannot exceed the present.

2.5.3. Non-functional test

Non-functional testing focusses on validating the way the system works and not through specific behaviors.

The purpose is to ensure that everything works well and to know in what circumstances they might fail.

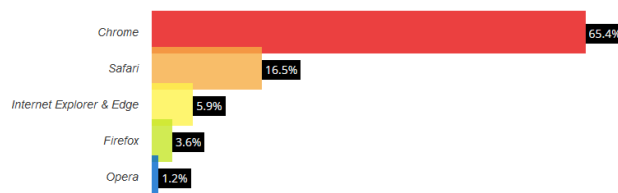
Scope: The below excel sheet details at high level the scope of Non-functional Test

Method: The tests will be automated tests with Selenium/Selenium WebDriver. The execution of the automated tests will be carried out after bug fixed issues as regression testing.

Test Acceptance Criteria:

- The form must be accessible by at least the four main browsers in their latest versions:

1. Chrome
2. Safari
3. Edge
4. Firefox



- The form must be usable by at least the most common resolution of each device:
 1. Desktop display 1920x1080
 2. Laptop 1024x 768
 3. Tablet 1280x800
 4. Mobile 640x360



TestCasesListxlsx.xls

x

NOTE: When the JIRA management tool is in use, this import of test cases will be automatic

2.6. Bug Triage

The severity and priority of new bug will be evaluated and prioritized by Testing Team and reviewed with the development team leader. If necessary, the product manager would be included in the review process.

Once a consensus has been agreed on a defect assessment, the bug reporter will assign the agreed resolution and the developer team leader will choose the member of his team in charge of fixing it.

2.7. Test Completeness

QA Complete is reached when the following conditions have been met:

- Test coverage has reached 100%

- All manual and automated tests have been executed
- All reported bugs are resolved or assigned for the next version.

2.8. Test Deliverables

No.	Deliverable Name	Author	Reviewer
1	Test Plan	Test Lead	Product Manager/ Project Manager
2	Test Cases	Test Team	Project Manager
3	Bugs Report	Test Team	Test Lead/ Developer lead
4	Weekly status report	Test Lead	Project Manager
5	QA Complete report	Test Lead	Project Manager

3. EXECUTION STRATEGY

3.1. Test Cycles

- There will be two cycles for functional testing. Each cycle will execute all the test.
- The objective of the first cycle is to identify any blocking, critical bugs, and most of the major issues.
- The objective of the second cycle is to identify remaining critical and major bugs, update test and obtain status results.

3.2. Validation and Bug Management

- The bugs will be tracked through bugs reports (To be managed and tracked with the JIRA tool in the future).
- It is the responsibility of the tester to open the bugs, perform traceability with the test and assign an initial severity and status.
- The tester has the responsibility of retest and close, if it is applicable, the reported bugs.
- Bugs reported during the testing will be categorized according below categories:

Severity	Impact
Blocking	This bug makes the system not work and it is not possible to test it
Critical	Bug capable of triggering complete system shutdown
Major	Bug capable of collapsing large parts of the system
Minor	Results in some unexpected or undesired behavior, but not enough to disrupt system function
Low	Bug won't result in any noticeable breakdown of the system

4. TEST MANAGEMENT PROCESS

4.1. Test Management Tool

- All testing artifacts such as Test cases, test results are updated in “LFR-Form-TestCases” excel file (To be managed and tracked with the JIRA tool in the future).
- Each feature and scenario will have its own unique identifier.
- Each Tester will directly access their respective assigned test cases and update the status of each executed step.
- Any defect encountered will be raised by a bug report linking to the particular Test Case.
- Various reports would be generated from “LFR-Form-TestCases” excel file information such as Test Report, Bugs Report or Status Report.

4.2. Test Design Process

- The tester will understand each requirement and prepare corresponding test case to ensure all requirements are covered.
- Each Test case will be mapped to Features as part of Traceability matrix.
- Each of the Test cases will undergo review by the Product.

4.3. Test Execution Process

- Once all Test cases are approved and the test environment is ready for testing, tester will start an exploratory test of the application to ensure the application is stable for testing.
- Each tester performs step by step execution and updates the executions status.
- If any of the tests fails, the bug must be linked in the excel management file with the test case failed and registered in the Bug Report.
- Testing team will participate in defect triage meetings in order to ensure all test cases are executed with either pass/fail category.
- Once the bug has been marked as resolved, the execution of the test case against the branch of the repository in which it is solved will be repeated.
- If a bug with resolved status is tested and the test case passes, the bug will be marked closed.
- This process is repeated until all test cases are executed fully with Pass/Fail status.

4.4. Test Risks and Mitigation Factors

Risk	Prob.	Impact	Mitigation Plan
Schedule: If the start of the testing is delayed due to design tasks.	High	High	The test team will follow the preparation of the tasks and will be involved from the analysis itself to have an early communication with the development team.
Resources: There are not enough resources and the onboarding process takes	Medium	High	<ul style="list-style-type: none">• Holidays are estimated within the calendar.• The experience of the team members

too long.			<p>(Expert, Senior, Junior, Scholar) will be taken into account when estimating the duration of the tasks within the schedule.</p> <ul style="list-style-type: none"> • Being 50% more hours of work for each lower category; 1h of Expert, 1'5h of Senior, 2'25h of Junior, ...
Bugs: Detecting bugs at a late stage in the cycle can lead to a longer resolution time.	Medium	High	<ul style="list-style-type: none"> • This is usually caused by an unclear specification. • The review of test cases and features must be reviewed in parallel by all parties involved. • Bug management must be fluid to ensure immediate communication and fixing of issues.
Scope: Scope completely defined before closing the definition of functionalities.	Medium	Medium	Do not close the definition of the scope until the requirements are well specified or take into account their update before changes.
Non-availability of Test environment and accessibility	Medium	Hight	Leave a gap of tasks that do not depend on the availability of the environment to advance in the schedule.
Testing delays due to new issues	Medium	Hight	<p>During the testing phase it is common for bugs to lead to new functionalities. It is important to start the testing phase as soon as possible so that the appearance of new issues does not affect too much the development already finished.</p>

5. RESOURCE AND ENVIRONMENT

5.1. Testing Tools

- Microsoft Office or equivalent to develop Plans, Reports and Bugs Tracking.
- IntelliJ IDEA 2021.2.3 with Gherkins and Cucumber for Java plugins installed.
- Selenium/Selenium Webdriver as JAVA library.
- Git Bash console installed in local.
- GitHub repository.
- Dockers.
- Jenkins LTS image for Dockers.

5.2. Test Environment

- The reports and plans will be generated with a text document tool.
- The list of Features and Test Cases will be managed by means of an Excel file.
- The definition of automatic tests will be done in the IntelliJ IDEA development environment using the Gherkin Domain Specific Language (DSL).
- We will use the Cucumber tool for the execution of functional descriptions in plain text with Gherkin language.
- The functionality of the Steps will be developed in JAVA language.
- Selenium for JAVA as a web-based test environment.
- Automated test development should be separated into logical parts in a new branch of our local repository.
- Changes to a local branch will be committed through a pull request to the remote repository.
- We will use the Docker Compose of the repository to mount a Docker image locally with Jenkins LTS.
- We will generate a Job in Jenkins against our remote repository that is in charge of checking the new requests that arrive and making sure that the compilation of that branch is correct before it merges with the “main” branch.
- We will generate a second Job in Jenkins against our repository that is in charge of executing the automated tests.

END OF THE DOCUMENT