Seismic - Test Plan

Objective of the document

This document outlines the overall planning and related information around the QA activities during the project execution. As per the scope defined in SoW, the HZTL QA Team will perform the tests listed below during the appropriate sprints. Both functional and non-functional tests derived as per the scope will be tested to validate CMS Implementation, Front-End UI and the Performance of the application, as per the requirements and agreements specified. The document also has information around the required pre-requisites that can allow safe start and continued execution of their assignments. Failure to any of these pre-requisites, might delay in delivery of QA assignments.

Pre-requisites

The pre-requisites listed below will allow QA to kick-off and continue their designated tasks without any hurdles.

- QA Team members allocated on the project will have access to Test Management and Project Management systems before their work starts
- Approved User Acceptance Criteria which clarifies the functional scope and expectations of the client, will be defined in JIRA/Project Management tool
- Stories based on these Acceptance Criteria will be created by relevant team members like BA or TL or PM which can be assigned to the team members further.
- Relevant access to VPN, RDP, or any other systems for QA(s)
- Access to required testing tools like LambdaTest, SortSite, any extensions required, on the client environment/systems
- Test Environment setup and access to QAs, ready for QA
- Approved Functional specs and Designs (at least for the Stories on which the team is going to work on)

⚠ **Kindly note that if any of the pre-requisites are not taken care of, it *might* further delay in expected progress and deliveries of the QA assignments.**

QA Process

Based on the Acceptance Criteria defined…

- User Stories will be created in Jira or equivalent Project management tool
- QA will be create Test Cases in QMetry (or other Test Management tool as stated in document) for the Functional and Non-Functional Stories created in Jira or relevant project management tool as mentioned.
- Test case executions will be performed for Stories assigned to QA for verification on a dedicated QA Environment. The status of tickets will be updated accordingly.
- Bugs will be created for deviations found against the primary source of truth/reference as decided and documented in the system
- Bugs can be triaged to PM/TL/GD if in case there is a difference of opinion. Upon agreement next steps can be taken.
- Bugs with lesser priorities, due to time constraint or other reasons, can be moved to future sprints, upon mutual team agreement and consent of the client.
- The QA verified and signed off code can be moved to a higher environment (UAT Environment) for the client to perform User Acceptance Testing (UAT), based on the Acceptance Criteria defined.
- Deviations found by Client needs to be logged in Jira with required information to reproduce the same on QA environment, so that it can be fixed and retested.
- UAT can be performed within the sprints as well as at the end of the project development, preferably after the actual Content is populated, to identify real-time defects.

Types of Testing

As a part of QA Capabilities, we can perform a lot of different types of tests relevant to the project needs. Please refer the types of testing listed below along with the information on whether that testing type is a part of this project or not.

Functional Testing

The checks below are generic ones applicable to both types of applications, Web and Mobile.

	Types of Testing	In Scope?

1	**Build Verification Testing**	**Yes**
2	**Integration Testing**	**Yes**
3	**Regression Testing**	**Yes**
4	**System Testing**	**Yes**
5	**In-Sprint UAT**	**Yes**
6	**Cross Browser and Device Testing**	**Yes**
7	**API Testing**	
8	**Multi-Lingual Testing**	
9	**Localization Testing**	
10	**Globalization Testing**	

Mobile App specific checks

The list of checks below are specific to Mobile applications.

	Type of Testing	In Scope?
1	**Notification Testing**	
2	**Battery Performance Testing**	
3	**Application Install/Uninstall/Update Testing**	
4	**Network Connectivity Testing**	
5	**Permission Testing**	
6	**Session Testing**	
7	**Cross Platform Testing**	
8	**Interruption Testing**	

Other Testing types

	Type of Testing	In Scope?
1	**Client UAT**	Yes, with the approach defined in the section below
2	**Accessibility Testing**	Yes, with the scope and approach defined in the section below
3	**Performance Testing (Define Scope if yes)**	Yes, with the scope and approach defined in the section below
4	**Automation Testing**	

Client UAT

Pre-requisites

- Well defined, detailed acceptance Criteria should be documented, approved and shared with the team during the start of the project.
- User Stories to be derived based on Acceptance Criteria, to ensure optimal implementation coverage
- Any changes in the requirements during Implementation phase, should reflect the Acceptance Criteria and hence the User Stories. Team needs to be updated on any such change ensuring the change is not impacting the timelines and schedule.

UAT Execution

- UAT Session Kick-off date, Session schedule, and duration should be well defined, giving the team sufficient time to perform retesting for any deviations found.
- A BA, PM or a Tech-lead ideally owns the UAT demos.
- Other team members including and not limited to QAs and Content Authors, at least one representative from each department working on the project, needs to be present in the demo to respond to or note down information as needed.
- Specific time needs to be allotted for the UAT sessions to all team members that will need to remain present in the sessions.
- It is recommended to perform the checks using the actual data. Client needs to provide the actual data before scheduling the UAT sessions.
- If actual data / content is not ready with the client, any post Content Authoring observations to be fixed *can be* re-evaluated whether they are an easy fix or will need time. This might lead to a separate *Change Order* or can lead to change / delay in project release dates / timelines. A discussion and an agreement need to be done upon mutual discussion.
- Acceptance criteria to be considered as baseline
- Selection of the browser and device combination should be aligned with the Browser and Device standards agreed in the SOW.
- Observations which were not a part of the requirements, *can be* considered as Change requests. An agreement between HZTL and the client is important to avoid misunderstandings.

Accessibility Testing

Horizontal will develop work outputs striving for WCAG 2.1 AA level compliance, although no specific compliance level can be guaranteed since significant aspects of accessibility compliance (such as UX/UI design and content) fall outside Horizontal’s control and are instead Client responsibilities. Any accessibility testing and verification done through the development sprint process will only provide partial verification of accessibility conformance. Manual accessibility testing and testing with assistive technologies, like Screen Readers, is not in scope. Additionally, Horizontal is not responsible for any accessibility issues through content or content authoring created or completed by the client, nor for any issues that can only be detected through manual and assistive technologies testing. For manual testing and screen reader verification testing, we recommend third party partnerships like Allyant who specialize in performing manual accessibility and screen reader testing.

Performance Testing

Performance testing is crucial for ensuring a positive end user experience. It assesses speed, responsiveness, and stability under diverse conditions. At Horizontal, our default testing offering includes a total of three (3) executions of Load testing for up to ten (10) static pages: two (2) Load test executions with Average User Load and one (1) Load test execution with Peak User Load, accommodating a maximum of three hundred (300) Virtual Users per test execution. Additionally, we provide five (5) dedicated IPs for whitelisting in restricted environments as part of our default testing offering. Using tools like JMeter for scripting and BlazeMeter, or a tool of Horizontal’s choosing similar to BlazeMeter, for execution, Horizontal simulates real-world scenarios to identify potential performance issues, enhancing system reliability.

Additional Notes:

- If the client desires adjustments to page counts, user counts, or other configurations beyond Horizontal's default testing offering, or if other types of tests (Stress, Spike, and/or Soak/Endurance) are desired, these customizations necessitate estimation based on requirements analysis. This process enables Horizontal to determine the appropriate tests to conduct and the associated additional costs, which may require a mutually agreeable and signed written Change Order.
- By default, performance and load test execution by Horizontal will be performed either with Horizontal's BlazeMeter or a tool of Horizontal's choosing similar to BlazeMeter, and a performance testing tool license and usage fee (typically $1,500) will be included in the costs of this project. If Client owns and has a working instance of BlazeMeter and permits Horizontal to use Client's BlazeMeter instance, this license and usage fee can be removed.
- Client should check with Client's chosen hosting provider(s) (such as Vercel, Netlify, etc.) to ensure that pre-go-live performance testing will not result in unexpected hosting charges. Horizontal can advise and assist with this communication upon request.

Examples of out-of-scope testing factors (non-exhaustive list) include the following for all performance testing types:

- Features/Component like Multi-Factor Authentication, OTP-based logins, ReCAPTCHA, Payment Gateways, etc.

Scope of Testing

Based on the discussions, communications and reference documents below is the list of Components and Functionalities that will be tested by the assigned QA for this project. However, during the project execution, upon discussion with client and as agreed by HZTL, these components and functionalities might change.

Please find the list of components and features below:

1. Sample Component 1
2. Sample Component 2

Out of Scope: Anything and everything that is not defined as a part of the QA scope above, needs to be considered Out of QA Scope

Defect Lifecycle

As a default workflow and configuration in JIRA, below are the statuses set for a Bug. The meaning of each status is noted below. However, there are fair chances that the project management tool is configured in a way that it has more statuses. The purpose of the same needs to be made clear with the team before they start using it.

	Bug Status	Details
1	**New**	A new bug is logged and is yet to be reviewed by Development team
2	**Active**	The bug is under review by the development team. In case of bug fix is not working then QA team will update bug status to Active
3	**Resolved**	The bug is resolved, and code fix is also deployed to respective environment, and it is ready for QA team for retesting
4	**Closed**	The bug is fixed. Or the bug no longer reproduces

Environment Details

QA environment

- The QA environment will be used for testing and validation by HTZL QA team. The first round of testing will be performed on this environment.

UAT environment

- This Test environment will be used for testing and validation by Client UAT team. It is designed to match production as closely as possible.

Production environment

- The Production environment is responsible for serving the latest versions of software, products, or updates are pushed into live, usable operation for the intended end users.

Story Status Classifications

Mostly when there is a ticketing system like Jira, there are different types of ticket statuses that come into picture. It is important to know their significance.

1. **New**
    - The user story is created and added to the backlog.
    - Requirements and acceptance criteria are defined but not yet approved.
    - QA may review and provide early feedback on the scope and testability.
2. **In Dev**
    - Development work has started based on the defined acceptance criteria.
    - Unit tests may be written and executed by developers.
    - QA can begin preparing test cases, setting up test environments, and identifying test data requirements.
3. **Ready for Deployment**
    - Development is complete, and the build is prepared for deployment to the testing environment.
    - Code reviews and internal checks have been completed.
    - QA ensures that all necessary test environments and dependencies are ready.
4. **Ready for QA**
    - The build has been successfully deployed to the QA environment.
    - QA verifies that all required components, integrations, and test data are available.
    - Smoke testing is performed to ensure basic functionality before proceeding with full testing.
5. **In QA**
    - QA executes functional, regression, and exploratory tests.
    - Any defects identified are logged, and the story may be sent back to **In Dev** for fixes if necessary.
    - If no major defects are found, the story progresses to the next stage.
6. **Closed**
    - The story has passed all required testing, including regression and UAT (if applicable).
    - QA has provided sign-off, and the story is considered complete.
    - The feature is released to production or ready for business use.

QA Involvement Across Story Statuses

- **New:** Early requirement analysis and test case planning.
- **In Dev:** Test case creation, environment preparation, and reviewing acceptance criteria.
- **Ready for Deployment:** Ensuring environments are set up and pre-validation checks are completed.

- **Ready for QA:** Performing smoke testing before starting formal test execution.
- **In QA:** Executing tests, logging defects, and ensuring fixes are verified.
- **Closed:** Providing final validation and sign-off, ensuring release readiness.

By following these clearly defined statuses, QA ensures a structured testing workflow, improves defect tracking, and enhances overall software quality.

Defect Classification

Priority

- **Priority 1:** This generally occurs in cases when an entire functionality is blocked, and testing cannot proceed because of this. Or in certain other cases if there are significant memory leaks, then generally the defect is classified as a priority -1 meaning the program/ feature is unusable in the current state.
- **Priority 2:** Priority 2 level bugs typically refer to issues that are important but not critical. These bugs can impact usability, performance, or functionality, but usually have workarounds or acceptable alternatives.
- **Priority 3:** Priority 3 level bugs in software testing are typically classified as lower priority issues. These bugs do not significantly impact core functionality or usability but still need to be addressed at some point.
- **Priority 4:** Priority 4 level bugs in software testing are the lowest priority issues. These bugs are often considered minor, trivial or cosmetic in nature, with minimal impact on functionality, usability, performance, or overall user experience.

Severity

- **Critical:** A defect which has high business impact like revenue loss, brand damage, etc. then the defect could be classified under critical severity.
- **Major:** Any Major feature implemented that is not meeting its requirements/use case(s) and behaves differently than expected, it can be classified under Major Severity.
- **Medium:** Any feature implemented that is not meeting its requirements/use case(s) and behaves differently than expected but the impact is negligible to some extent, or it doesn’t have a major impact on the application, can be classified under Medium Severity.
- **Low:** Any cosmetic defects including spelling mistakes or alignment issues, or font casing can be classified under Low Severity

Entry | Exit | Suspension Criteria

In a test plan, entry, exit, and suspension criteria are essential to define when testing should start, stop, or be temporarily halted. A common criteria that will be applied across wherever applicable is defined below:

Entry Criteria

Entry criteria are the conditions that must be met before testing begins. Project team should make sure that the entry criteria is met before the tickets are assigned to QA for testing.

- **Requirements are defined and approved:** Test requirements and specifications must be clearly documented and approved by stakeholders.
- **Test environment is set up:** The testing environment, including hardware, software, network configurations, and any required test data, must be ready and validated.
- **Test cases are prepared:** All test cases, scripts, and necessary data should be written, reviewed, and approved.
- **Dependencies are resolved:** All dependencies, such as tools, software versions, and interfaces, must be available and functioning correctly.

- **Test tools are configured:** Any testing tools (e.g., automation frameworks, defect tracking systems) should be installed, configured, and tested.
- **Initial code or system build is available:** The build or code to be tested must be available and meet the necessary quality standards (e.g., passes smoke testing).

Exit Criteria

Exit criteria define when testing can be considered complete.

- **All planned test cases executed:** All test cases planned for the testing phase have been executed.
- **Defects resolved:** All critical and major defects have been identified, reported, and resolved or deferred with agreement from stakeholders.
- **Defect rate within acceptable limits:** The number and severity of open defects should be with no showstoppers or critical defects remaining.
- **Performance benchmarks met:** The application/system meets performance benchmarks (e.g., response time, throughput) as per the requirements.
- **Stakeholder approval:** If the client is going to be a part of Sprint level UAT, their sign-off on the stories becomes mandatory for the team.
- **Documentation completed:** All test results, logs, defect reports, and test summaries are documented and archived.

Suspension Criteria

Suspension criteria outline the conditions under which testing should be temporarily halted.

- **Environment issues:** Testing environment is unavailable or unstable, impacting the ability to execute tests.
- **High defect rates:** A high number of critical defects or blockers are found, preventing further testing until they are resolved.
- **Incomplete deliverables:** Key deliverables like test data, code, or necessary documentation are not ready or incomplete.
- **Critical resource unavailability:** Key personnel (e.g., testers, developers, or SMEs) required for the testing process are unavailable.
- **System crashes:** Frequent system crashes or instability issues make it impossible to proceed with testing.
- **Performance issues:** Severe performance degradation prevents the continuation of tests.

Tools Considered for this project

Function	Tool used	Configuration, license or other related details
Project Management / Ticketing system	HZTL JIRA	
Test Management system	QMetry integrated with HZTL JIRA	
Bug Tracking system	HZTL JIRA	
Repository of Information/Documentation	HZTL Jira / Confluence	
Designs provided in	Figma	
Tool for Cross Browser / Device testing	LambdaTest	

Browser and Device Details

QA team will be using the browsers and devices listed below for the specific tests

Web Application Testing

List of Browsers on Desktop

	Desktop Operating System	Browsers	Browser Version	Screen Resolution	Tested on
1	Windows 10	Google Chrome	Latest	1920 X 1080	Local system
2		Microsoft Edge	Latest	1920 X 1080	Local system
3	Mac (Ventura)	Safari	Latest	1920 X 1080	LambdaTest

List of Browsers on Mobile devices

	Devices	Operating System	Browser
1	iPhone 14	iOS (Latest supported on LambdaTest)	Safari (Latest)
2	Samsung Galaxy S22	Android (Latest supported on LambdaTest)	Chrome (Latest)

Note: The **screen resolution** will be a **default one, specific to the devices** listed above using ***LambdaTes*t** as applicable.

Mobile Application Testing

	Mobile App Devices	Operating System	Orientation	Tested on
1	iPhone 14	iOS	Portrait	Physical device
2	Vivo Y21 or Samsung A14	Android	Portrait	Physical device

Deliverables

As a part of documentation and deliverables, QA team will share the documents below with the relevant stakeholders. Please guide them on the cadence and recipients of the relevant documents.

	Document Name/Type	Schedule of document to be shared to stakeholders
1	**Test Plan Document**	Start of the project. To be approved by the listed team members to avoid gaps
2	**Test Cases**	End of each Sprint
3	**Test Execution Report**	End of each Sprint
4	**Bug Report**	End of each Sprint
5	**Automation Test Scripts**	As per agreement between HZTL and client, at the end of the project.

6	**Accessibility Testing Report(s)**	Exported from the tool used to perform checks
7	**Performance Testing Report**	Tool generated report to be shared after each execution, along with the basic observations.

Team and Document details

Team info

Project details	
Project Name:	
Test Plan Created By:	
QA Team members on Project:	
Team member 1:	
Team member 2:	
Test Plan Creation Date:	

Reviewer Info

Reviewers		
	Reviewer Name	**Review Date**
Project Manager:		
Tech Lead:		
QA Lead:		
Client Representative:		

References

- Statistics and their snapshots are from StatCounter
- Responsive testing can happen on local system or using LambdaTest.

[End of document]