**Test Plan**

Contents

1. [Introduction](#page3)

2. [Document Objective](#page3)

3. [Scope](#page3)

4. [References](#page3)

5. [Testing Process Overview](#page3)

6. [Testing Strategy](#page4)

7. [Tools](#page4)

8. [Test Environment](#page4)

9. [Test Schedule](#page5)

10. [Deliverable](#page5)

11. [Entry Criteria](#page5)

12. [Suspension Criteria](#page5)

13. [Resumption Criteria](#page5)

14. [Exit Criteria](#page6)

15. [Risk](#page6)

1. Introduction

Web accessibility refers to the inclusive practice of removing barriers that prevent interaction with, or access to websites, by people with disabilities. When sites are correctly designed, developed and edited, all users have equal access to information and functionality.

2. Document Objective

Objective of this Test plan is to define the testing strategies which we are going to use throughout the process and testing tools used for complete testing life cycle of this project.

3. Scope

The website will be tested by its functionality and design against accessibility for all types users. All requirements will divide into modules and for every delivery milestone features will be tested with the complete flow for that delivery.

1. References
   * Test Cases prepared by QA Owner and approved by QA Lead.
   * Material/ soft copy of requirements provided by client/Team Lead.
2. Testing Process Overview

Testing process followed by QA will be categorized into two ways:

* Process to be followed when sufficient time is available for QA.

o Understand complete project requirements and raise queries if any.

o QA will prepare test cases based on the requirement specifications. This will cover all modules and scenarios of all requirements.

o Test cases will be executed by the respective QA member.

1. Retesting of the fixed bugs will be done by respective QA once it is resolved and bug/defect/suggestion's status will be updated accordingly. In certain

cases, regression testing will be done if required.

1. Once all the outstanding bugs/defects are fixed by the developer, code will be deployed. If there is any urgency to deploy the code on client's server before clearing outstanding bugs, System Analyst needs to formally approve it to QA.
2. One round of testing will be done by QA on client's test environment if required. High level test cases will be executed after the release on production.

* Process to be followed when sufficient time is not available for QA:

o Understand the requirements and raise the query if any.

o QA will be doing Ad-hoc testing based on requirements and test scenarios.

1. Retesting of fixed bug will be done by respective QA once it is resolved by respective developer and bug/defect will be updated accordingly. In certain cases, regression testing will be done if required.
2. Once all the outstanding bugs/defects are fixed by the developer, code will be deployed to the client's test it by developer.
   1. One round of testing will be done by QA on client's test environment if required
3. Testing Strategy

For testing the entire system, QA will follow following types of testing.

* + **Functional Testing**: Functional testing is carried out in order to find outunexpected behavior the front end and the back end. The characteristic of functional testing are to provide correctness, reliability and accuracy of the system.
  + **GUI Testing**: GUI testing will include testing the UI part of the system. It will covertest against mock ups, error messages, spelling mistakes etc. Mock ups will also be tested under this as per the requirement.
  + **Responsive Testing**: Front end website will include the responsiveness on Mobile,tablet and desktop.
  + **User Acceptance Testing:** The purpose behind user acceptance testing is toconfirm that system is deployed according to the specified user requirements and is ready for operational use.

1. Tools
   * **Google sheet** will be used for Test cases
   * Devices to verify the **responsive** will be decided as per the scope of project.
2. Test Environment
   * Staging server will be used to test the build. After the complete verification when all/major bugs will be resolved a stable build will be deployed.
   * Once UAT will be done, system will be deployed on production server and a round of testing will be done on Production server

9. Test Schedule

The Key milestone regarding testing is shared below in the table for the project. Milestone dates can be modified except for external deliveries. Any changes in schedule dates must also be highlighted to the QA team.

|  |  |
| --- | --- |
| Key Milestone | Target Date |
| Release 1 | Date(dd-mm-yyyy) |
| Release 2 |  |
| Release (n) |  |

10. Deliverable

The key QA documents deliverable for the project is shared below

|  |  |
| --- | --- |
| Deliverable | Responsibility |
| **Test Plan** | QA will provide the test cases which will be further verified |
|  | by QA Lead |
|  |  |
| **Test Cases** | QA will provide the test plan which will be verified by QA |
|  | Lead. |
|  |  |
| **Bug Report** | Bug report will be created for all milestones according to the |
|  | requirement otherwise bug details will include in QA report itself. |
|  |  |

1. Entry Criteria
   * The whole source code must be unit tested based on test cases & test scenarios.
   * QA resource have completely understood the requirements/ functionality.
   * Test scenarios and test cases must be reviewed by QA Lead
2. Suspension Criteria
   * The build contains many serious defects which seriously or limit testing progress.
   * Significant change in requirements suggested by client/Team Lead.
   * Software/ Hardware problems.
   * Assigned resources are not available when needed by test team.
3. Resumption Criteria

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

14. Exit Criteria

- No defects over a period of time.

- All the high priority/ severity test cases have been executed.

- Deliverable are ready.

- High Severity/Priority bugs are fixed or discussed in bug triage.

15. Risk

- Communication gap between what is built and what was to be built.