|  |  |  |
| --- | --- | --- |
| **Prepared By** | **Prepared On** | **Version No** |
| Kierson Vigilla | 10/2023 | 2023-0001 |
|  |  |  |

1.0 INTRODUCTION

This document describes the overall testing plan for the Pagefreezer’s Public Portal Website. The test plan specifies the objective, scope, and testing strategy for the site. It also includes the estimates on resource allocation and testing effort for the website.

2.0 OBJECTIVE

The objective of the test is to verify the common functionalities of Pagefreezer’s Public Portal website if they are performing in accordance with the specifications. The tests execute the test cases and record the test results either Passed or Failed. At the end of testing, it concludes the quality of the website and be used as one of the criteria in releasing the software.

3.0 SCOPES

The scope of this test includes the functionalities of the following:

* Test if the Public Portal Website is accessible.
* Test if Search is working.
  + User can type search keyword test & use the filters to get the results using “Advanced Search”.
* Verify both Social Media & Websites Tabs.
* Some Negative Tests:
  + Blank Search Criteria
  + Blank Date Range in advanced search
  + Switched Date Range

The scope of this test does not include functionalities outside the *openrecords/automation* endpoint.

4.0 FEATURES

* Search
* Advanced Search
* Social Madia Tab
* Websites Tab
* Sort by
* Filter by

5.0 TEST STRATEGY

Functional testing of the features will be executed using both manual and automation. Automated testing will run using Java and Selenium under Firefox browser. Manual testing will be performed on Chrome browser.

Entry criteria for test execution is when the migration of testable codes is deployed in the test environment. Codes shall be comprised with the features specified by the stakeholders, and the fixes for the defects/bugs raised during the development phase.

Note: No further deployments should be made once the test execution starts. If such event occurs, test execution will be reset or will have a targeted testing.

Exit criteria is when the features and bug fixes are marked as Passed:

- 100% Tests executed.

- Test results for all tests are captured and documented.

- 95% Tests get the Passed status.

- No outstanding defect with high severity.

- All defects during test execution are documented.

- Low severity defects are marked as invalid or deferred for the next release.

Suspension criteria is when the number of issues found during testing comprise at least 50% of the test cases. Testing may also be suspended if there are impending connection issues with the server of the test environment where the release candidate software is deployed.

Defect management is categorized according to the severity of the issue being affected. Defect is captured and discussed on the next daily stand-up meeting to mitigate and perform the necessary next steps. This is captured using the bug reporting tool with the following categories of severity:

Critical - Defects which causes the system to crash. It may also cause the system to return unexpected errors or errors that are not handled properly by the codes.

High - Defects which affects the normal flow of the program and does not have a workaround for the user to navigate through. It may also be a showstopper for any related feature.

Medium - Defects which affects the quality of the system but with workaround for the user to navigate through. User will still be able to complete the workflow of the system.

Low - Defects which has a minor or no heavy impact on the workflow of the system. It may consist of trivial behavior of some web elements or UI issue which affects the aesthetics of the website.

In the event where a defect is encountered relating to a feature or module being developed, the related ticket of the feature will be linked to the defect ticket. Defect tickets will have supporting steps on how to recreate the defect and a screenshot to show which page/feature it is affecting.

Defect testing/retesting is assigned back to the QA who initially reported the defect to retest the fixes and provide the relevant test results.

Each QA directly assesses the assigned feature to be developed. Test cases will be updated or created accordingly based on the description of the feature indicated on the ticket/requirement. Created test cases will be available to all the development team members. Created test cases will be reviewed by the Business Analyst to check whether the scenarios correspond to the requirements. Any updates on the test cases will directly be updated and will still be available to all development team members.

Each QA performs test execution assigned to them and will update the test case with the test results accordingly. Test results will comprise of the actual result or behavior of the feature under test and a screenshot to show evidence of the behavior. QA will also indicate either the test is Passed or Failed. Tickets will be reopened and assigned back to the Developer who worked on the feature if there is an issue with the behavior or if the actual test results are not the same with the expected test results.

QA will have defect triage meetings every after sprint grooming in order to assess both outstanding and deferred defects. This will also ensure that all reported defects are discussed and mitigated before performing regression testing.

6.0 RISK ASSUMPTIONS

Itemized below are the risks of the project which affects the testing procedures or the execution of the test plan:

- Tight Deadline: Delayed deployment of release candidate software in the test environment will create a bottleneck for QAs. Testing cannot be extended due to the UAT schedule in place.

Mitigation: QA will add buffer to the schedule and prepare the test data which can be added before upgrading the test environment or before deploying the release candidate.

- Resource Allocation: QAs handling multiple projects may take time before getting on board or may allocate part-time only for certain projects when doing integration or regression testing.

Mitigation: Projects will have certain QAs who will be the main tester. They will coordinate the scope to other QAs when preparing for the integration testing or regression testing. They will also be readily available to devote full-time on the assigned project once the integration testing or regression testing started.

- Late Defects: Defects which are uncovered at the last stages of the testing will most likely be deferred on the next sprint, except for those which poses a great impact on the business.

Mitigation: Plan for managing raised defects will be observed and followed in order to communicate the defects accordingly to the development team.

- Test Environment Server: Issues on the server handling the test environment is not stable or if the server is not readily available at the start of the testing phase.

Mitigation: Development team will reach out to middleware team and give a heads up to do the necessary steps in checking the stability of the connection on the server. Middleware team should be available to assist if there comes a time where the server becomes unstable during testing.

7.0 SCHEDULES

Day1 - Day2: Release candidate deployment and Test Data creation

Day3 - Day8: Test Execution (Manual and Automation)

Day9 - Day10: Test reporting and defect triage/mitigation

Day11 - Day12: Test Delivery and sign off

8.0 RESOURCES

QA Manager - Coordinate testing schedules and resource allocation.

QA Specialists - Perform manual testing, log defects and coordinate test results.

Developer in Test - Perform functional testing using automation.

Test Administrator - Middleware team member who will be the contact person for the server.

9.0 TOOLS

Laptops/Desktops running on Windows OS.

Chrome and MS Edge browsers for manual test execution.

Firefox for automated test execution.

Separate test environment and server for running test automation.

10.0 TEST DELIVERABLES

Test Report containing the summary of the test run and the logged defects.

Test Cases containing test results with screenshots as evidence.

Test Automation run result with error logs and execution logs.

Itemized defect report.

Test plan document.

Test sign-off.