Weboin Test Plan

Created by: Shobhit Verma
Doc no. : Test Plan WEBOIN v1.0

List of contents

Overview

Purpose

This test plan describes the testing approach and overall framework that will drive the testing of the weboin.

Scope

This document details the testing that will be performed by the testing team for the weboin project. It defines the overall testing requirements and provides an integrated view of the project test activities. Its purpose is to document:
· What will be tested;
· How testing will be performed;
· What resources are needed, and when

Objective

The objective of this test plan is to verify that the functionality of weboin works as per the specifications

Test strategy

Test strategy will help clients to understand what type of testing the test team is going to perform and how it's going to perform. Every type of test contains objective, Approach, Activities, Exit criteria which will give a clear idea.

1. Functional Test

This test is done to verify that the functions of application work as per the specifications and all the expected workflow and requirements are being met if viewed for the user's eye.

1a.Objective of Functional test:

This test is to verify that the functions of application work as per the specifications. Testing will be focused on meeting the business objectives and quality. There will be common, consistent procedures for all teams supporting the testing activities. The testing Processes will be well defined yet flexible with the ability to be changed if needed. The testing activities will build upon previous stages to avoid duplication of efforts. The testing environment and data will emulate a production environment as much as possible. It will be a repeatable and measurable process. It will be divided into distinct phases, each with clearly defined objectives. There'll be an entry and exit criteria.

1b. Approach of functional test

Functional testing of weboin will contain preloaded test data which Is available on the system during the test execution, used for testing activities. This functional testing is carried out by

feeding input and validating the output from the application. The testing will be performed by the testing team.
There will be 2 cycles for functional testing. Each cycle will execute all the scripts. The Objective of the first cycle is to identify and block critical defects, and most of the high defects. It is expected to do some work around in order to get all the scripts. The objective of the second cycle is to identify remaining high and medium defects, remove the work around from the first cycle, Correct gaps in the scripts and obtain the performance result.

2c. Activities performed in functional test:

1. Determine which functionality of the product needs to be tested. This can vary from testing main functions, messages, error conditions and/or product usability.
2.Create input data for functionalities to be tested according to specified requirements.
3. Determine acceptable output parameters according to specified requirements.
4. Execute test cases.
5. Compare actual output from the test with the predetermined output values. This reveals if the system is working as expected.

It is expected that the testers will execute all the relevant test cases in each of the cycles mentioned in the approach section. They could also do additional testing if they find gaps in the scripts. If a gap is identified, the scripts and the traceability matrix will be updated and then a defect will be logged against the test case. It is the responsibility of the tester to create the defect, link them to the corresponding test case and assign a severity of defects and facilitate with the technical team to get a fix and its implementation and communicate with the team members to continue or to halt the testing process. Defects logged during testing will be categorised according to the bug reporting tool google sheets used for test management.

1. Each resource in the testing team will be provided with read/write access to add/modify test cases in the google sheet, so that every change to the test cases can be updated directly in the click up.
2. Any defect encountered will be reported in the click up.
3. The tester verifies the defect fix and updates the states in click up.

2d. Exit criteria in functional test:

1. Execution of all the functional test cases has been completed.
2. No critical or P1, P2 bugs are open.
3. Reported bugs have been acknowledged.

2.System Test

2a. Objective of system test:

One of the primary objectives of System testing is to reduce risk. Even after individual testing of components, the risk of how they will all come together to form a complete System still exists. System testing eliminates this risk by ensuring that it will function as per customer requirements.

1. System testing must verify whether the design of the functional and non-functional behaviours of the system is as per the customer's specifications.
2. Validate that the system is complete and will work as expected.
3. System testing aims to build confidence in the quality of the system as a whole.
4. System testing also aims to find defects and to prevent defects from escaping to higher test levels or production. Additionally, it is the only phase that occurs on the full System just before the User Acceptance testing. So, it's critical to find all the possible defects at this stage, and they don't leak to production.
5. System Testing results are used by stakeholders to make release decisions. The Entry criteria for User Acceptance testing is the basis completion of System Testing. System testing may also adhere to legal or regulatory requirements or standards.

2b. Approach of system test:

It is mainly a Black-box type testing. This testing evaluates the working of the system from a user point of view, with the help of a specification document. It does not require any internal knowledge of systems like the design or structure of the code.
It contains functional and non-functional areas of application/product.
Focus criteria:
It mainly focuses on the following:

1. External interfaces
2. Multiprogram and complex functionalities
3. Security
4. Recovery
5. Performance
6. Operator and user’s smooth interaction with the system
7. Installability
8. Documentation
9. Usability

2c. Activities of system test:

Given below are the various activities involved while performing this testing:

- The very first step is to create a Test Plan.
- Create System Test Cases and test scripts.
- Prepare the test data required for this testing.
- Execute the system test cases and script.
- Report the bugs. Re-testing the bugs once fixed.
- Regression testing to verify the impact of the change in the code.
- Repetition of the testing cycle until the system is ready to be deployed.
- Sign off from the testing team.

d. Exit criteria for system test

- No critical or Priority or security-related bugs should be in an open state.
- All the test cases should be executed.

- If any medium or low priority bugs are in an open state, then it should be implemented with the acceptance of the customer.
- Exit Report should be submitted.

3. Performance Test

Performance test will be conducted to ensure that the application response time meets the user expectations under heavy stress and/or volume. Since this application can be used by a wide number of users we need to include a performance test to ensure system stability in high stress situations.

3a. Objective of performance test:

The objective of performance testing is to evaluate the performance and scalability of a system or application under various loads and conditions. It helps identify bottlenecks, measure system performance, and ensure that the system can handle the expected number of users or transactions. It also helps to ensure that the system is reliable, stable and can handle the expected load in a production environment.

3b. Approach of performance test:

Performance Testing simulates the real-world user load on the server and identifies the performance of the system in terms of:
- Users Load
- Response Time
- Request Rate

A performance tester conducts the performance testing with the help of a specific testing tool known as Performance Testing Tool.

3c. Activities performed in performance test:

1. Gathering the requirements.
2. Selection of tools.
3. Performance testing will be performed using that tool.
4. Test result analysis.
5. Generating reports.
6. In the report if there is a bug that will be fixed by the dev team.
7. After the fix is done the tester will retest it and complete the performance testing.

3d. Exit criteria of performance test:

1. Tests have been run.
2. System performance has been assessed according to the goals of the testing.
3. Test report written and passed to relevant people.
4. Fault reports written for any issues.
5. Performance goals have been achieved - only applies if the testers are working with the developers to (iteratively) test and tune the system until it meets the goals.

6. Need to derive a benchmark for performance which will be used for evaluation in future.

4.Security Test

Security testing will determine the security of the application. The test will verify that the unauthorised user access to confidential data is prevented.

4a. Objective of security test

Security Testing is a type of Software Testing that uncovers vulnerabilities of the system and determines that the data and resources of the system are protected from possible intruders. It ensures that the software system and application are free from any threats or risks that can cause a loss. Security testing of any system is focused on finding all possible loopholes and weaknesses of the system

4b. Approach of security test

1. The first step is to understand the business requirements, security goals, and objectives in terms of the security compliance of the organisation. The test planning should consider all security factors.
2. Understand and analyse the requirements of the application under test.
3. Collect all system setup information used for the development of Software and Networks like Operating Systems, technology, hardware. Make out the list of Vulnerabilities and Security Risks.
4.Reports: Prepare a detailed report of Security Testing which contains Vulnerabilities and Threats contained, detailing risks, and still open issues etc.

4c. Activities performed in the security test:

1. Requirement stage: Security Procedures In the requirement phase of SDLC, we will do the security analysis of the business needs and also verify which cases are manipulative and malicious.
2. Development or coding stage: Security Procedures: In the coding phase of SDLC, we will perform the white box testing along with static and dynamic testing.
3. Testing (functional testing, integration testing, system testing) stage:
Security Procedures: In the testing phase of SDLC, we will do one round of vulnerability scanning along with black-box testing.
4. Implementation stage:
Security Procedures: In the implementation phase of SDLC, we will perform vulnerability scanning again.
6. Maintenance stage
Security Procedures: In the Maintenance phase of SDLC, we will do the **impact** analysis of impact areas.

4d.Exit Criteria for security test:

All the security bugs should be fixed, retested and closed. Make sure that weboin application should be fully tested in terms of security.

5. Regression Test

Regression will be performed on areas of the system that previously had major defects and to make sure the changes made in the application did not cause any adverse effect on the system.

5a. Objective of regression testing

The first objective in bug-fix testing is to check whether the bug fixing has worked or not. Therefore, you run exactly the same test that was executed when the problem was first found. If the program fails on this testing, it means the bug has not been fixed correctly and there is no need to do any regression testing further. Finds other related bugs It may be possible that the developer has fixed only the symptoms of the reported bugs without fixing the underlying bug. Moreover, there may be various ways to produce that bug. Therefore, regression tests are necessary to validate that the system does not have any related bugs.

5b. Approach of regression testing:

- Measure or change the scale of the upgrade to determine how likely it would affect the application.
- When introducing automation Api testing, outline the test scripts to know which should be automated.
- The regression testing approach must cover all the possible test cases and impacted functionalities.
- Focus on the testing process and technology
- Perform risk analysis based on the size of your business/project and its complexity, along with its importance

5c. Activities of regression testing:

Ideally, you perform regression testing every time your code base is modified, whether through an update, a fix or other changes. Some organisations perform it every time a change is pushed. After the code is debugged and defects are fixed, take the resources and timeline into account to inform the regression testing approach. Choose one of the types mentioned above, then progress through the following steps:

5c1. Gather test cases

If you plan to select particular test cases, the first step is to identify those tests. Some areas of the application that might require attention include main functionality, error-prone segments of code and complicated user sequences. If comprehensive written test cases do not already exist, take the time to document them so that they can reliably be executed again and again.

5c2. Prioritise test cases

Even if you plan ahead, you might not have time to test it all. Development delays and rushed delivery timelines can reduce the amount of testing time remaining. Keep productivity in mind from the beginning by prioritising test cases. You might prioritise according to the functionality with the highest business impact, which features present the most risk or another criteria. You can also classify tests for future use to avoid delay in future iterations of this step.

5c3. Run tests

Execute these tests using the solutions of your choice, keeping in mind the work schedule, release schedule and ongoing feature development your teams may need to support.

5c4. Defect logging

Log all the bugs in the bug management tool.

5c5. Retest the bug

When fixes are completed retest the bugs and other modules which can be affected while fixing the following bug.

5d. Exit criteria of regression testing

- All P1 and P2 bugs should be fixed after regression testing..
- There is a chance that 5% of P2 bugs from the 100% bug list are not solved and 15% for P3.

6. Cross Browser Test and Multiplatform Test:

6a. Objective of Cross Browser Test and Multiplatform Test:

This test will be done to make sure the application is compatible with different browsers and platforms.

6b. Approach of Cross Browser Test and Multiplatform Test:

Choosing the right combination of browsers and devices for cross browser testing is nothing less of a giant leap covering a major milestone in the ongoing process. Using your product’s usage data to determine which devices and browsers your customers prefer is a key component to start the process. Once you have calculated and analysed the types of devices and browsers your customers prefer, you will need to revise the list to meet the changing browsers and devices in the target market. Once the list of prioritised and must have browsers and devices are made, you will need to make sure that you check and take action when older devices and browsers disappear, and newer ones appear.

6c. Activities performed in the Cross Browser test and multiplatform Test:

6c1. Establish a baseline: Before you begin cross browser testing, run all the design and functionality tests on your primary browser-usually Chrome. This will give you an idea of how the website was originally intended to look and behave.

6c2. Create a testing plan and pick the browsers to test on: Use the test specification document to outline exactly what you'll test. Then, as outlined in the segment above, pick browser-OS combinations to test based on popularity and site traffic analysis.

6c3. Execution: Execute test scenarios in the selected browsers and note the defects if they exist.

d. Exit criteria:

1. weboin application should be working properly in all the selected browsers.
2. All issues logged should be resolved.

7. User Acceptance Testing (UAT)

will be performed once the application is ready for implementation. The purpose of this test is to confirm the system is developed as per the specified user requirement and is ready for operational use.

7a. Objective of user acceptance test:

UAT is testing the integration of a computer system into a much larger system called the business or organisation. It is a form of Interface Testing and is concerned with checking communication between the system and the users. This does not mean it is a form of Usability testing, which checks how easy it is to work with a computer system. Instead, it is about whether a business or organisation can input the information they need to and get back usable results which will enable the business to go forward.

7b. Approach of User acceptance testing:

1. Analyse product requirements and define key deliverables.
2. Choose the approach of user testing.
3. Implement end-user testing tools.
4. Create the user acceptance environment and run training.
5. Run the tests.
6. Log bugs and when it is fixed retest it.

7c. Activities perform in user environment testing:

There are a series of steps to follow to prepare for UAT:

1. Prepare formal and informal scripts for business users - use your use cases to build test scripts that will serve as training materials for users on how to use the system after deployment
2. Use a test management tool - put your scripts in a tool and teach users how to use it
3. Create master data for business users to use for testing - organise this data by test case so users can easily reference what data they should use for each script. Include logins, passwords, and data they must look at or consume in the tool
4. Create a UAT user manual or guide - include where to access logins, URLs, and master data
5. Teach business users how to write a good defect - use a defect tracking system and teach users what information to include (like steps to recreate) and, if appropriate, how to set priority values
6. Coordinate the build schedule with your developers - ensure your dev team is aware of the testing schedule, so they don't do a build while users try to test. Ideally, you coordinate with the entire IT team

7d. Exit criteria for user acceptance testing:

Following are the exit criteria for User Acceptance Testing:
1. No showstoppers, high & medium defects open
2. Smooth working of business process
3. UAT Sign off meeting with stakeholders

Tools used for testing:

Below mentioned tools will be used.
1. Google sheet: This tool will help us to manage the Test scenarios and Test runs and the most important thing is bug tracking.

Test Environment:

For the test environment, a key area to set up includes
1. System and applications
2. Test data
3. Database server
4. Front-end running environment
5. Browser
6. Hardware includes Server Operating system
7. Network
8.Documentation reference documents, configuration guides,
Installation guides, user manuals.
9. Setting up all the tools in the system. which is mentioned in the tool used for the testing section in the system.

Functions to be tested:

1. Functions to be tested in Customer portal

a. Auth (Login Registration)
b. Customer Profile
c. Personalization
d. user Listing & Search
e. Create post
f. Create reel
g. Create stories
h. Trending section
I. filters
j. Blocking functionality and listing
k. Saved post functionality
l. Delete account functionality.
m. like, share and comment section.

Stakeholders lists

These are the people involved in the planning, management, and execution of the application **weboin**
1. Project Manager Reviews the content of the Test plan, Test strategy and Test estimates and signs off on it.
2. Business Analyst develops the software requirement specifications (SRS) based on the client's needs.
3. Development Lead serves as a primary contact/liaison between the development team and the project team.
4.The Development Team develops the software. They generate all the programming or the code for the software. This team fixes defects according to the schedules.
5.Testing Team develops Test conditions, Test cases, executing scripts and Expected results.

Deliverables

Below Dates are the sample dates of project

Completion Date	Responsibility	Deliverables

Suspension/ Exit criteria

If any defects are found which seriously impact the testing progress, the QA lead may choose to suspend testing criteria that will justify test suspensions are:
1. Hardware/Software is not available at the times indicated in the project schedule.
2. Source code contains one or more Critical defects, which seriously limit the testing process.
3. Assigned test resources are not available when needed by the team.

Resumption Criteria

If testing is suspended, resumption will occur when the problems that caused the suspension have been resolved. When a critical defect is the cause of the suspension, the “FIX” must be verified by the test Department before testing is resumed.

Dependencies

1. Software dependencies: The source code must be unit tested and provided within the scheduled time outlined in the project schedule.
2. Hardware dependencies: A proper working pc’s should be available for every member of the testing team.
3. Test Data & Database: Test Data & Database should also be available to the testers for use during testing.

Test Report

Our project development team is using google sheet for project management and test management tools. So with the help of this tool. The report will be documented in such a way that it will give a proper understanding of how many test cases are planned, executed, passed and failed and Defect Summary-Severity wise; Defect Distribution-Function/Module wise; Defect Ageing etc..

Please find the google sheet link in which test management is handled.

Click here Weboin test management