TEST PLAN

Prepared by: Inaam Ullah

Company: Ilsainteractive

AUG 25, 2022

Version History:

Version	Revised By	Summary	Approval	Date

Table of Contents

13. Tools and defect Tracking

14. Final Test Report

15. Exit Criteria

1. Introduction

The Test Plan outlines the scope, approach, resources, and schedule of all testing activities. It identifies the items and features to be tested; types of testing. It contains a detailed and executable strategy for conducting. It defines the detailed testing objective specific to a particular system, the testing approach, test environment, test conditions, and the test plan.

2. Scope

The scope of this test plan is to ensure websiteX meets all of its technical, functional and business requirements. The purpose of this document is to describe the overall test plan and strategy for testing the website. The approach described in this document provides the framework for all the testing related to website. This document will also be updated as required with the requirement updates. We also need to make sure that all the expected results are achieved.

2.1. Out of Scope

3. Test Objectives

The general test objectives are to test the correctness of the generation of the interface data file, the content of the interface data file, and any error conditions. The quality objectives of testing the website are to ensure complete validation of the business and software requirements:

- Verify software requirements are complete and accurate
- Perform detailed test planning
- Identify testing standards and procedures that will be used while testing the website
- Prepare and document test scenarios and test cases

- Manage defect tracking process
- Finalize the prject for release

4. Reference Documents

_--------

5. Detailed Test Approach

Detialed testing phases and methodologies are mentioned below. We will follow the protocols of each phase and achieve the highest results.

- **Requirement Analysis**
- **Design Testing**
- **Functionality Testing**
    - **Verify each functionlaity of the system is working as per requirement**
    - **Testing the links**
    - **Testing the forms**
    - **Cookies Testing**
    - **Validation (HTML/CSS/PHP)**
    - **Database Testing**
- **Integration Test Specification**
- **API Testing**
- **Usability Testing**
- **Compatibility Testing**
    - **Browser Compatibility**
    - **OS compatibility**
    - **Mobile browsing**
- **Performance Testing**
    - **Load Testing**
    - **Stress Testing**
- **Security Testing**
- **Automation Testing**
- **Smoke Testing**
- **Beta Testing**

5.1 Requirement Analysis

Requirements analysis is critical to the success or failure of a systems or software project so we have to verify, validate and confirm each requirement. Requirements must be

validated on the basis of User Experience, User Interface, How to test the requirements, Requiremnts are up to date.

Make sure the major scenarios and requirements are mentioned in the document. If there is something missing, highlight the missing requiremnts and also suggest improvements if there is any.

Analyze the following Requirements

—----------

5.2 Design Testing

Test all the designs and verify all the designs must be correct as per the requirements.

And also make sure the designs for all the specified languages and dark themes.

5.3 Functionality Testing

Test the functional requirements and determine every function of the software is acting in accordance with the pre-determined requirements and tasks. At website, Performed testing of all the links in web pages, checking the database connections, forms used in the web pages for submitting or getting information from user & Cookie testing. Functional testing is extended to the types given below.

5.3.1 Testing all the Links:

1. Test the outgoing links from all the pages from a specific domain under test.
2. Test all internal links.
3. Test links jumping on the same pages.
4. Test links used to send the email to admin or other users from web pages.
5. Test to check if there are any orphan pages.
6. Lastly in link checking, check for broken links in all above-mentioned links.

5.3.2 Testing of the forms on the web pages:

Forms are the essential and integral parts of website . Forms are used to get information from users and to keep interaction with them.

The following should be checked on the forms:

- Check all the validations on each field.
- Check for the default values of fields.

- Wrong inputs to the fields in the forms.
- Options to create forms, if any, form delete, view or modify the forms.
- Check that no empty forms are created.
- There are different field validations like email-id's, user financial
- information, date, etc
- All the above validations should be checked in a manual or an
- automated way.

5.3.3 Cookie Testing:

Cookies are small files stored on a user machine that are basically used to maintain the sessions such as the 'login sessions'. Test the website to verify

- Test the application by enabling or disabling the cookies in your
- browser options
- Test if the cookies are encrypted before writing to user machine
- During the test for session cookies (i.e. cookies expire after the
- sessions ends) check for login sessions and user stats after session end
- Check effect on application security by deleting the cookies

5.3.4 Validation (HTML/CSS):

HTML/CSS validation is very important for optimizing the website for

search engines.

- The site has full and correct Doctype
- The site uses character set
- The site uses valid XHTML
- The site uses valid CSS
- The site has no unnecessary ids or classes
- The site uses well structured code
- The site has no broken links
- The site has clearly defined visited links

5.3.5 Database Testing:

- Check for data integrity and errors while you edit, delete, modify the forms or do any database related functionality.
- Check if all the database queries are executing correctly and data is retrieved correctly and also updated correctly.

Also we need to validate the Database by executing the queries.

Comp

5.4. API Testing

Set of procedures to verify the expected functionality, reliability, and security and ensure the correct interaction between backend and frontend. To validate the logic of the build architecture within a short amount of time. Each api test consists of some test actions mentioned below. Further details of API Testing will be covered in API Test Plan

- Verify the URL accordingly environment
- Verify required request headers and their correct values
- Verify response payload
- Verify correct HTTP status code and response headers
- Verify expected result and correct application state
- Verify correct performance sanity

5.6. Usability Testing:

The goals of usability testing include establishing a baseline of user performance, establishing and validating user performance measures, and identifying potential design concerns to be addressed in order to improve the efficiency, productivity, and end-user satisfaction.

The usability test objectives are:

- To determine design inconsistencies and usability problem areas within the user interface and content areas.
- **Potential sources of error may include:**
    - Navigation errors – failure to locate functions, excessive keystrokes to complete a function, failure to follow recommended screen flow.
    - Presentation errors – failure to locate and properly act upon desired information in screens, selection errors due to labeling ambiguities.
    - Control usage problems – improper toolbar or entry field usage.
- Exercise the application or web site under controlled test conditions with representative users. Data will be used to access whether usability goals regarding an effective, efficient, and well-received user interface have been achieved.
- Establish baseline user performance and user-satisfaction levels of the use interface for future usability evaluations.
- **Basic Usability:**
    - The site should have a clear hierarchy.
    - Headings clearly indicate the structure of the document
    - Navigation should be easy to understand

- Navigation is consistent throughout the site
- The site uses underlined links
- The site uses consistent and appropriate language
- The site has easy to find sitemap and contact page
- The site has a search tool
- The site has a link to home page on every page
- The site has clearly defined visited links

5.7. Compatibility Testing:

5.7.1 Browser compatibility

Some requirements are very dependent on browsers. Different browsers have different configurations and settings that the web page should be compatible with. The web site coding should be cross browser platform compatible. Test the UI of website, functionality, security checks or validations then stresses browser compatibility testing of the web application. Test web application on different browsers like Internet explorer, Chrome, Firefox, Netscape navigator, AOL, Safari, Opera browsers with different versions

5.7.2 OS compatibility:

Some functionality in the web application may not be compatible with all operating systems. All new technologies used in web development like graphics designs, interface calls like different API's may not be available in all Operating Systems. Testing the web application on different operating systems like Windows, Unix, MAC, Linux, Solaris with different OS flavors.

5.7.3. Mobile browsing:

Mobile browsing is also an integral part of browsing. Testing the web pages on mobile browsers is highly important. Compatibility issues may be there on mobile. Currently ,the system is not designed for mobile browsing, although this is an area we can add in future.

5.8. Performance testing:

Website should sustain heavy load. website performance testing should include: Load Testing & Web Stress Testing. Test the website on different Internet connection speeds.

5.8.1. Load Testing:

Test website if many users are accessing or requesting the same page. Can the system sustain peak load times? Site should handle many simultaneous user requests, large input

data from users, Simultaneous connection to DB, heavy load on specific pages etc. For load testing we will use blazemeter and robot scripts.

5.8.2. Stress Testing:

Generally stress means stretching the system beyond its specification limits. Web stress testing is performed to break the site by giving stress and check how the system reacts to stress and how the system recovers from crashes. Stress is generally given on input fields, login and sign-up areas. In website performance testing website functionality on different operating systems, different hardware platforms are checked for software, hardware memory leakage errors. For stress testing we will use jmeter and blazemeter.

5.9. Security Testing:

- Test by pasting the internal URL directly into the browser address bar without login. Internal pages should not open.
- If you are logged in using a username and password and browsing internal pages then try changing URL options directly. I.e. while checking website Try directly changing the URL site ID parameter to a different site ID, which is not related to, logged in user. Access should be denied for this user to view others' stats.
- Try some invalid inputs in input fields like login username, password, and input text boxes. Check the system reaction on all invalid inputs.
- Web directories or files should not be accessible directly unless given
- download option.
- Inspect the pages and verify Elements, Console and Sources.
- Test the CAPTCHA for automated script logins.
- Test if SSL is used for security measures. If used proper message should
- get displayed when the user switches from non-secure http:// pages to secure https:// pages and vice versa.
- All transactions, error messages, security breach attempts should get logged-in log files somewhere on the web server.
- Check the security points while the communication between frontend and backend.
- It may includes:

    User IP

    Accept-Language

    SESSION_ID/Device ID

    Signature

    User-Agent

Furthermore we will use Zap for security testing. It is an end-to-end web application security scanner. This will give us a 360-degree view of the security of our website. It is important to have an understanding of how the client (browser) and the server communicate using HTTP. the tester should at least know the basics of SQL injection and XSS.

Automation Testing:

5.10. Automation Testing:

Automate all the implemented functinlaities of the website . Creation of the automation test cases. Details of Automation testing will be discussed and covered inside the Test Plan of automation testing.

- Write the test cases in a test rail for specific features.
- Analysis of test cases which are possible to be automated
- Plan the test cases and how to execute all the test cases.
- Write the test cases w.r.t environment.
- write the logic for how to validate the test.
- Write the pass/fail criteria agaist each test case.
- run and actually verify the test.
- Integrate all the test cases of each feature/module

5.11. Smoke Testing:

Smoke Testing is a software testing process that determines whether the deployed software build is stable or not. Insid the smoke testing QA Engineer will make sure all the critical functionalities are working fine. We will create a checklist of smoke testing.

Smoke testing will be performed at two stages. Once new features are added, the other is before finalizing the build for Production/live.

Create the checklist for smoke testing.

5.12. Beta Testing

Beta testing is basically a relase for specific users to use a product in a production environment to uncover any bugs or issues before a general release.

Beta testing is the final round of testing before releasing a product to a wide audience. The objective is to uncover as many bugs or usability issues as possible in this controlled setting. QA will also a perfrom the beta Testing.

6. Test Strategy

The overall strategy of this testing initiative is manual, black box testing. We are testing the data, interface part and mplemented system in detail. The testing at the SAP end of the interface will be covered by the SAP functional testing. Follow the testing phases and techniques mentioned inside **“Detailed Test Approach”.** All type of testing are covered in this document.

Some of the test specifications use test data which needs to be set-up in the test environment prior to executing the test cases.

For each level of testing, a separate test plan is prepared with the following set of deliverables:

- Test Cases/Test Sceanrios
- Features to be tested
- Items to be tested
- Pass / Fail criteria
- Bugs cycle
- Automation
- Expected Results
- Actual Results

7. Test Schdeule

The test schedule is the timeline of acceptance testing activities and deliverable dates.

Testing activities are mentioned below.

- Requirement Analysis.
- Design Testing
- Develop test scenarios
- Develop test cases
- Review scenarios/test cases for accuracy, completeness and sequence (confirm test data is correct)
- Integration testing
- API Testing
- Regression Testing
- Functionality Testing
- Database Testing
- Integration Test Specification
- Usability Testing

- Compatibility Testing
- Performance Testing
- Security Testing
- UAT Testing
- Automation Testing
- Smoke Testing
- Beta Testing

8. Problem/Bug Severity Classification:

The identified severity for each problem implies a general reward for resolving it, and a general risk for not addressing it, in the current release.

Severity 1 - Crash or High impact problems that often prevent a user/host from correctly completing an experience/booking.

Severity 2 - Moderate to high frequency problems with the functionality/UI or UX impact

Severity 3 - Either moderate problems with low frequency or low problems with moderate frequency; these are minor annoyance problems faced by a number of participants.

Severity 4 - Low impact problems faced by few participants; there is low risk of not resolving these problems. Reward for resolution is typically exhibited in increased user satisfaction.

9. Test Resources

Here is the list of resources with the roles those will work on website

1.
2.

10. Pass/Fail Criteria:

Create the test cases and mention the expected results/pass criteria against each testcase.

11. Environment

Start testing on a staging server once a certain level is achieved, then move to Production and give the final approval at Production. All the experiments should be performed at staging. Testing data must be private at Production.

12. Test Cases and Test Scenarios

Write down the detailed test cases on the basis of requirement, technical doecument and test plan. For testcases use the Google sheet and use the Jira for bug, suggestion reporting.

13. Tools and defect Tracking

Jira wil be used for defect reporting and issue bugs/defects management and traceability.

14. Final Test Report

Test closure reports shall be generated for each testing phase as the testing phase gets completed.

15. Exit Criteria

All the test cases and test scenarios must be passed. Every user must get the music recommendation as per their interests.