Syscosmic Technologies Pvt Ltd

Your Vision, Our Code. Elevating Success Together

Testing Guidelines for Web Application

Contents

Detailed Testing Guidelines for Web Application Testing

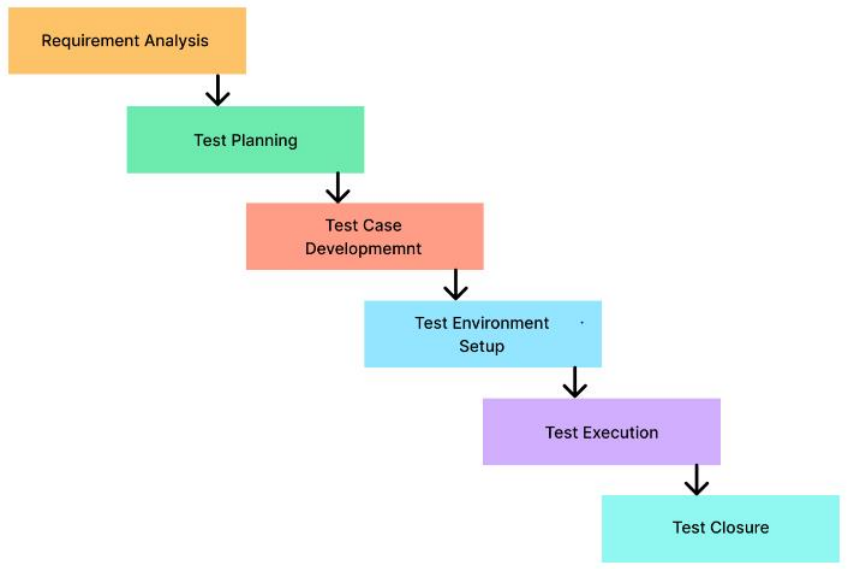
1. Introduction

Web application testing is crucial to ensure that a web application functions correctly, is secure, and provides a seamless user experience. This document outlines comprehensive guidelines for manually testing web applications, covering all key phases, types of testing, and best practices. It also lists free tools that can aid testers in conducting manual testing effectively.

---

2. Testing Phases

The web application testing process can be divided into several key phases:



2.1 Requirement Analysis

Objective: Understand the web application's scope, requirements, and objectives.

Activities:

Review project documentation, including functional and non-functional requirements.

Identify the critical features, modules, and areas that require special attention.

Gather information on the target audience, supported platforms, and expected usage scenarios.

Output: A clear understanding of what needs to be tested, including key focus areas.

2.2 Test Planning

Objective: Plan the testing process, including defining the scope, objectives, and resources required.

Activities:

Develop a detailed test plan document that includes testing objectives, strategies, scope, resources, schedule, and risk analysis.

Identify the testing environments, tools, and required skills.

Prioritize test cases based on critical functionalities and business impact.

Output: A test plan document that guides the entire testing process.

2.3 Test Design

Objective: Create detailed test cases and scenarios to be executed during testing.

Activities:

Design test cases based on the requirements, covering all possible user interactions and edge cases.

Develop test data and input values for various scenarios.

Review and validate test cases with stakeholders to ensure they align with the requirements.

Output: A set of detailed test cases ready for execution.

2.4 Test Execution

Objective: Execute the designed test cases and document the results.

Activities:

Perform manual testing by executing the test cases.

Record the actual outcomes and compare them with the expected results.

Identify and document any discrepancies, defects, or unexpected behavior.

Output: Test execution results, including a list of identified issues.

2.5 Defect Reporting

Objective: Document and communicate any defects or issues found during testing.

Activities:

Log defects in a defect tracking system with detailed descriptions, steps to reproduce, severity, and screenshots if applicable.

Assign defects to the appropriate team members for resolution.

Retest the resolved issues and update the defect status accordingly.

Output: A defect log that tracks the status of all identified issues.

2.6 Test Closure

Objective: Conclude the testing phase and ensure all critical issues are resolved.

Activities:

Review the test execution results to ensure all critical test cases have passed.

Conduct a final round of regression testing to confirm that recent changes haven't introduced new issues.

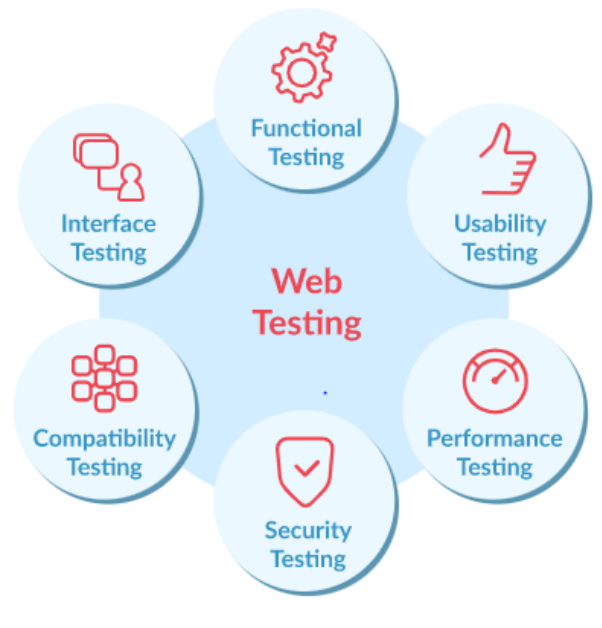
Prepare a test summary report detailing the overall test results, including pass/fail rates, defect statistics, and key findings.

Output: A test closure report that confirms the web application is ready for production.

---

3. Types of Testing

To ensure the web application's overall quality, various types of testing should be performed:



3.1 Functional Testing

Objective: Ensure that all web application features work as intended according to the requirements.

Activities:

Test all user interactions, such as clicking buttons, submitting forms, and navigating through different modules.

Validate that input fields accept and process data correctly.

Verify that business logic and workflows are implemented correctly.

Check the integration of third-party services, such as payment gateways, APIs, etc.

3.2 Usability Testing

Objective: Assess the web application’s user interface (UI) and user experience (UX) to ensure it is intuitive and user-friendly.

Activities:

Evaluate the ease of navigation and the logical flow of the application.

Test the readability of content, including font size, color contrast, and spacing.

Observe user interactions to identify potential usability issues, such as confusing layouts or unclear instructions.

Collect feedback from real users, if possible, to gain insights into the user experience.

3.3 Performance Testing

Objective: Test the web application's speed, responsiveness, and stability under various conditions.

Activities:

Conduct load testing to see how the application performs under high traffic or data load.

Monitor page load times and identify any performance bottlenecks.

Test the application's response time during peak usage periods.

Analyze server and database performance to ensure they can handle the expected load.

3.4 Compatibility Testing

Objective: Ensure the web application functions correctly across different browsers, devices, and operating systems.

Activities:

Test the application on multiple browsers, including Chrome, Firefox, Safari, Edge, and others.

Verify that the application renders correctly on various screen sizes and resolutions, including mobile devices, tablets, and desktops.

Test the application's functionality on different operating systems, such as Windows, macOS, iOS, and Android.

3.5 Security Testing

Objective: Identify and address potential security vulnerabilities in the web application.

Activities:

Test for common security threats like SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), etc.

Validate that sensitive data, such as passwords and personal information, is properly encrypted and stored.

Check for secure communication protocols (e.g., HTTPS) and proper session management.

Ensure that user authentication and authorization mechanisms are robust and secure.

3.6 Accessibility Testing

Objective: Ensure the web application is accessible to users with disabilities and complies with accessibility standards.

Activities:

Validate that the application meets the Web Content Accessibility Guidelines (WCAG) standards.

Test the application's compatibility with screen readers and other assistive technologies.

Check for appropriate use of alternative text for images, keyboard navigation support, and color contrast.

Ensure that forms and other interactive elements are accessible and easy to use for all users.

4. Manual Testing Guidelines

Manual testing is essential for assessing the web application's usability, functionality, and overall user experience.

Project Name:	Surgery App	Test Designed by:	Nikhil Nigam				Testing status			
Project Cordinator	Anurag Sir	Test Designed date:	04/24/2024				Pass	6		
Test Priority	end to end testing	Web link	https://surgeryapptest.azurewebsites.net/Admin/Dashboard				FAIL	42		
							Fixed	0		
							Enhacement	1		

Testcase number	Role	Credential	Module/Control	Testing Method	Testcase criteria	Test Steps	Expected result	Actual Result	Testcase result	Tester Remark/ATTACHMENT
	host	id-admin password-123456	hospital management/organization list	functional	check user is able to see created organization in organization list and add, delete and get login url action perform or not	1) ckick on the given link https://surgeryapptest.azurewebsites.net/Admin/Dashboard 2) enter login credentials 3)select 3 lines or menu select hospital management /organization/ organization list	user should be able to see created organization name on oorganization list forma and user should be easily perform add delete and edit and get login url action perform	user is able to see created organization name on oorganization list forma and user is easily perform add delete and edit and get login url action perform	Pass	
	host	id-admin password-123456	hospital management/organization list/i button	functional	check in dark theme , all saved data is correctly shown or not on i button click or not	1) ckick on the given link https://surgeryapptest.azurewebsites.net/Admin/Dashboard 2) enter login credentials 3)select 3 lines or menu 4)select hospital management /organization/ organization list/i button	user should be able to see data clearly on dark theme	on i button click in organization list form , data is not clearly visible in dark theme	Fail	attachment

---

4.1 Prepare Test Cases

Objective: Develop detailed test cases that cover all functionalities and edge cases.

Activities:

Write test cases that clearly describe the steps to be performed and the expected outcomes.

Ensure test cases cover all user interactions, including positive, negative, and edge cases.

Organize test cases by functionality or module for easy execution.

Output: A comprehensive set of test cases ready for execution.

4.2 Perform Exploratory Testing

Objective: Discover unexpected issues by testing without predefined test cases.

Activities:

Explore the web application freely, trying different combinations of actions and inputs.

Test unusual or edge case scenarios that may not be covered by standard test cases.

Use your intuition and experience to identify potential problem areas.

Output: A list of any issues discovered during exploratory testing.

4.3 Check Cross-Browser Compatibility

Objective: Ensure the web application works correctly on different web browsers.

Activities:

Execute test cases on various browsers like Chrome, Firefox, Safari, Edge, and Opera.

Identify and document any browser-specific issues, such as layout problems or JavaScript errors.

Output: A compatibility report highlighting any browser-related issues.

4.4 Test Responsiveness

Objective: Verify that the web application adapts properly to different screen sizes and devices.

Activities:

Test the application on multiple devices, including desktops, tablets, and smartphones.

Ensure that the application's layout, text, and images adjust correctly across different screen resolutions.

Check the functionality of touch events, such as swiping and tapping, on mobile devices.

4.5 Validate Forms and Inputs

Objective: Ensure that all forms and input fields work as expected.

Activities:

Test form submissions with valid, invalid, and edge case data.

Check for proper validation messages and error handling.

Verify that data submitted through forms is correctly processed and stored.

Output: A report on the functionality of all forms and input fields.

4.6 Check Links and Navigation

Objective: Verify that all internal and external links, as well as navigation elements, are working correctly.

Activities:

Click on every link to ensure it navigates to the correct destination.

Test the functionality of navigation menus, buttons, and breadcrumbs.

Check for broken links and document any that return 404 errors or incorrect pages.

Output: A list of working and broken links, along with any navigation issues.

4.7 Review Content

Objective: Ensure that all text, images, and multimedia content are displayed correctly and without errors.

Activities:

Verify that all text content is free from spelling, grammar, and formatting issues.

Check that images and videos load correctly and are properly aligned.

Ensure that dynamic content, such as sliders and pop-ups, functions as intended.

Output: A report on the accuracy and quality of the web application's content.

4.8 Document Issues

Objective: Record any bugs or issues found during testing with detailed information.

Activities:

Log each issue with a clear description, steps to reproduce, expected and actual outcomes, and severity.

Include screenshots or video recordings if necessary to illustrate the issue.

Assign issues to the appropriate team members for resolution and track their progress.

Output: A detailed defect log.

5. Free Testing Tools for Manual Testing

5.1 Functional Testing Tools

A browser extension that allows you to create, record, and execute manual test cases directly in the browser.Ideal for testing complex user interactions.

Katalon Recorder:

A lightweight tool for recording and playing back test cases in the browser.

Useful for quickly creating tests without needing to write code.

5.2 Cross-Browser Testing Tools

In cross-device testing, we test an application across different devices. We ensure the applicatsion provides a smooth user experience and functions well on other devices like mobile phones, tablets, and desktops. So, no matter what device the user has, the application should behave consistently.

5.3 Accessibility Testing Tools

WAVE:

A web accessibility evaluation tool that helps identify accessibility issues on web pages.

Highlights elements that may be problematic for users with disabilities.

Axe:

A browser extension that provides automated accessibility testing within the browser.

Detects accessibility violations and suggests fixes.

5.4 Performance Testing Tools

Performance testing is the practice of evaluating how a system performs in terms of responsiveness and stability under a particular workload. Performance tests are typically executed to examine speed, robustness, reliability, and application size

---

5.5 Security Testing Tools

QA in Security Testing is crucial because it's like having a security guard for your digital world. It checks if your software is strong enough to keep hackers out. Without QA, hackers could sneak in and cause problems. With QA, we can find and fix any weaknesses before they become big issues

---

5.6 Mobile Testing Tools

LT Browser:

A tool specifically designed for testing mobile responsiveness.

Allows you to test your web application on various mobile devices and screen sizes.

Responsinator:

A simple online tool that shows how your web application looks on different devices.

Helps ensure your application is optimized for mobile viewing.

6. Best Practices

To ensure thorough and effective web application testing, consider the following best practices:

6.1 Keep Detailed Records

Objective: Maintain comprehensive documentation of all testing activities.

Activities:

Document test cases, test execution results, and defects in a centralized repository.

Keep a log of all testing activities, including the date, time, and results of each test.

Share test documentation with the development team to ensure transparency and collaboration.

Outcome: Clear and accessible records that can be referenced throughout the project.

6.2 Collaborate with the Team

Objective: Work closely with developers, designers, and other stakeholders to ensure issues are resolved promptly.

Activities:

Communicate regularly with the development team to discuss test findings and prioritize defect resolution.

Participate in stand-up meetings, sprint reviews, and retrospectives to stay aligned with the project goals.

Provide feedback on design and functionality from a testing perspective.

Outcome: A collaborative environment that fosters quick resolution of issues and continuous improvement.

6.3 Stay Updated

Objective: Keep up with the latest trends, tools, and techniques in web application testing.

Activities:

Regularly update your testing toolkit with the latest versions of tools and plugins.

Participate in webinars, workshops, and online courses to learn about new testing methodologies and technologies.

Stay informed about changes in web standards, accessibility guidelines, and security practices.

Outcome: A modern and effective testing approach that adapts to the evolving web landscape.

---

7. Conclusion

By following these detailed guidelines and utilizing the recommended tools, testers can ensure comprehensive and effective web application testing. This will lead to a more reliable, user-friendly, and high-performing web application. Consistent application of best practices, thorough documentation, and ongoing collaboration are key to successful web application quality assurance.