Theophilus onolunose

**PREPARED BY: Theophilus Onolunose**

**TEST PLAN WORDPRESS PROJECT**

**DATE: 16.09.2023**

**Table of Contents**

[Overview](#_bookmark0) [3](#_bookmark0)

[Scope](#_bookmark1) [3](#_bookmark1)

Features to be Tested [3](#_bookmark2)

[Out](#_bookmark3) of scope [4](#_bookmark3)

[Test Environments](#_bookmark3) [5](#_bookmark4)

[Test Strategy](#_bookmark5) [5](#_bookmark5)

[Structured](#_bookmark7) Test Workflow [5](#_bookmark7)

Best Practices for Enhanced Testing [5](#_bookmark6)

Defect Management Approach [6](#_bookmark8)

[Criteria](#_bookmark9) for Test Lifecycle Phases [6](#_bookmark9)

[Entry and Exit Criteria](#_bookmark11) [6](#_bookmark11)

[Risks and Mitigations](#_bookmark14) [7](#_bookmark14)

[Tools](#_bookmark13) [7](#_bookmark13)

**Overview**

As part of my project initiative, I am motivated to execute a comprehensive automated testing regimen for the web application located at 'https://wordpress.com/.

The product owner is 'WordPress’.

This document delineates the overarching test plan, encapsulating the project's scope, our strategic testing approach, timelines, resource allocations, and anticipated deliverables.

**Scope**

Within the framework of this testing initiative, our primary focus will be to rigorously evaluate specific functionalities of the 'https://wordpress.com/’ web application. Utilizing advanced automated testing methodologies, we aim to ensure the robustness and reliability of the platform. The project's ambit encompasses a thorough examination of key features, interfaces, and user interactions. This systematic approach will ensure that each component aligns with the predefined quality benchmarks, ensuring optimal user experience and platform stability.

Features to be Tested (Test items):

* Create Account
* Login & Logout
* Forgot Password
* Search and navigation
* Insert Products
* Uploads and Downloads Page
* Orders Page
* Add to Cart
* Product Display Page
* Wish List
* Shopping Cart
* “My store” Home Page
* Checkout Page
* My Account Page
* Contact Us Page
* Menu Options
* Multiple stores
* Category Pages
* Currencies

**Out of Scope**

Within the parameters of this automated testing initiative, the following elements are explicitly excluded:

* Features not delineated within the " Test items " section.
* Integrations pertaining to third-party functionalities, including payment gateways.
* The process of test automation itself, as the focus remains on the application's inherent functionalities and user interactions.

## **Test Environments**

* Windows 11 – Chrome, Firefox, and Edge
* Mac OS – Safari Browser

**Test Strategy**

With the ‘help’ section provided by 'WordPress' in their online platform, I have discerned the necessity for a comprehensive Functional Testing regimen tailored to the functionalities delineated in the "Test items" section. The strategic approach to this testing initiative encompasses:

* Test Design: Systematic development of test scenarios that encapsulate the diverse functionalities within the testing scope.
* Test Case Formulation: Rigorous creation of detailed test cases derived from the aforementioned scenarios, ensuring a holistic evaluation of each feature.
* Execution Strategy: Leveraging advanced automated testing tools and methodologies to ensure precision and efficiency in the testing efforts.

In the pursuit of ensuring comprehensive coverage and accuracy, the testing strategy will harness a blend of established and innovative Test Design techniques:

1. Structured Techniques:

1. Equivalence Class Partitioning: Segmenting input data into equivalent classes to minimize redundancy.
2. Boundary Value Analysis: Focusing on the edge values of input domains to identify potential anomalies.
3. Decision Table Testing: formulating test cases based on logical conditions and their corresponding outcomes.
4. State Transition Testing: Evaluating system behavior across different input conditions and transitions.
5. Use Case Testing: Simulating real-world scenarios to ensure end-to-end functionality.

2. Heuristic Techniques:

1. Error Guessing: Leveraging our expertise to predict potential error-prone areas and creating test cases accordingly.
2. Exploratory Testing: Empowering testers to dynamically explore and assess the application's behavior.

3. Test Case Prioritization: To optimize our testing efforts and ensure timely identification of critical issues, we will prioritize test cases based on their significance and potential impact.

**Structured Testing Workflow**

1. Preliminary Assessment: Upon receiving an application build, our initial step is to conduct a Smoke Test. This ensures that the core functionalities are operational, and the build is stable enough for further examination.
2. Build Acceptance: Should the Smoke Test reveal critical issues, we'll promptly reject the build, deferring comprehensive testing until a more stable iteration is provided.
3. Detailed Analysis: Once a stable build clears the Smoke Test, we delve into meticulous testing using our predefined test cases.
4. Parallel Testing: To ensure compatibility and consistency, multiple testers will evaluate the application across various supported environments concurrently.
5. Defect Management: Identified issues are logged in our defect tracking system. Additionally, a daily status report detailing the discovered defects is saved.

Testing Spectrum: The testing approach encompasses:

* Smoke and Sanity Testing: Initial assessment of core functionalities.
* Regression and Retesting: Ensuring modifications don't introduce new issues and verifying bug fixes.
* Usability, Functionality, and UI Testing: Evaluating user experience, feature completeness, and interface aesthetics.

**Best Practices for Enhanced Testing**

1. Context-Driven Testing: Our testing approach is tailored to the unique context and requirements of each application.
2. Shift-Left Testing: We initiate testing early in the development lifecycle, ensuring early detection and resolution of issues rather than awaiting a finalized build.
3. Exploratory Testing: Beyond structured test case execution, our team leverages their expertise for dynamic, uncovering potential edge cases and anomalies.
4. End-to-end workflow: Our approach will encompass evaluating the entire user experience, ensuring seamless integration and interaction across multiple functionalities, thereby replicating real-world user scenarios.

**Defect Management Approach:**

During Test Execution:

1. Anomaly Identification: Any deviation from the anticipated application behavior will be meticulously documented. If it doesn't qualify as a defect, it will be cataloged as an observation or raised as a query for further clarification.
2. Usability Assessment: Issues affecting the user experience or interface intuitiveness will be duly reported.
3. Defect Verification: Upon pinpointing a potential defect, its reproducibility will be verified. Detailed documentation, inclusive of screenshots and step-by-step reproduction procedures, will be maintained.
4. Daily Reporting: At the culmination of each testing day, a comprehensive report of identified defects, along with observations, will be saved for review.

Documentation:

1. Defect Tracking: All identified defects will be systematically recorded in an Excel spreadsheet and Xray integrating Jira, ensuring structured and accessible data management.
2. Test Documentation: Both test scenarios and test cases will be meticulously documented in a dedicated Excel workbook and Xray integrating Jira, facilitating clarity and traceability.

**Criteria for Test Lifecycle Phases:**

1. Requirement Analysis:

Entry Criteria:

The testing team is equipped with the official Requirement Documentation or detailed project specifications.

Exit Criteria:

The testing team has thoroughly dissected and comprehended the listed requirements.

All ambiguities or uncertainties have been addressed and resolved.

1. Test Planning:

Entry Criteria:

The team has distilled testable requirements from the provided documentation using the ‘help’ section provided in the webpage.

All queries stemming from the requirements have been clarified.

Exit Criteria:

The Test Plan, encompassing the overarching Test Strategy, has received the client's endorsement.

1. Test Designing:

Entry Criteria:

The Test Plan has been double-checked, ensuring alignment on testing objectives and approach.

Exit Criteria:

The Test Scenarios and Test Cases documentation has been meticulously formulated following the requirements provided in the ‘help’ section of the webpage.

1. Test Execution:

Entry Criteria:

The Test Scenarios and Test Cases have been double-checked.

The application is primed and set for rigorous testing.

Exit Criteria:

Comprehensive Test Case Reports and Defect Logs have been generated, detailing the testing outcomes.

1. Test Closure:

Entry Criteria:

Both Test Case Reports and Defect Logs are compiled and ready for review.

Exit Criteria:

A Test Summary Report has been curated, encapsulating the entirety of the testing process and its results.

# **Risks and Mitigations**

The following are the possible risks associated with the testing process and ways to mitigate them:

Risk: Captcha problem

Mitigation: Write a script to bypass it.

Risk: Non-Availability of a Resource

Mitigation: Backup Resource Planning

Risk: The wordpress.com is not working

Mitigation: It will work on other OS

Risk: Less time for Testing

Mitigation: Activate parallel testing

# **Test tools:**

The following are the list of Tools we will be using in this Project:

* Bug Tracking Tool
* Xray integrated into Jira
* Screenshot Tool
* Word and Excel documents