**Test Plan and Strategy Document**

Project Name : Automation Practice Ecommerce Website Testing Project

**Document Control:**

**Document Details:**

|  |  |
| --- | --- |
| Title | Test Plan and Strategy document for Automation Practice Ecommerce Website Testing Project |
| Version | 1.0 |
| Date | 6/23/2024 |
| File Name | <Document file name> |
| File Location | <Location of the file> |
| Author | Kirti Mhaske |
| Contributor |  |

**Change Control:**

|  |  |  |  |
| --- | --- | --- | --- |
| Issue Date | Version | Details | Author |
| <Issues date> |  | <Working Draft>, <Reviewed> | <Name> |
|  |  |  |  |

**Document Reference:**

|  |  |
| --- | --- |
| Document Name | File Location |
| <Document file name> | <Location of the file> |
|  |  |
|  |  |

# 1. Scope and overview:

# 1.1 Product Perspective

# The build for the e-commerce website is a new, self-contained product.

1.2 Product Functions

A user should be able to search for an item, and narrow down their search by selecting different filters for an item, they should be able to view details about an item, add an item to a cart, continue shopping, and then checkout when they are ready. In addition, there should be different filters for different types of items. In other words, items should be categorized into groupings of similar items. Furthermore, there should be search suggestions when a user starts typing, so that the correct category of item can be retrieved.

1.3 System Features and Characteristics

* New User Can Create an Account
* Returning User Can Login
* User Can Search For An Item
* User Can View Item Details
* User Can Add Item to Cart
* User Can View Their Cart
* User Can Checkout

1.4 Other Nonfunctional Requirements

1.4.1 **Performance Requirements**

New users should be able to create an account in less than 60 seconds. When a user searches for an item, they should receive a list of search results in less than 3 seconds. Once a user hits the “checkout” button, they should be able to complete the checkout form and receive a confirmation message in under 2 seconds. Speed is of utmost importance, as delays may cause a poor user experience and may deter users from returning to the site in the future.

1.4.2 **Security Requirements**

All PII should be protected and secure. Credentials should be verified before a user is given access to account information. In addition, passwords should be encrypted. Credit card /Bank related information should not be stored, and transactions should be processed securely.

1.5.4 **Software Quality Attributes**

The E-Commerce Website should follow web accessibility guidelines. In addition, the site should be easy to use and navigate. The website should be reliable, as interruptions in service may deter users from returning. The site should be scalable to accommodate for business growth and an increasing userbase. It is important that the site be modifiable, to allow for the addition of new features, and the product should be maintainable and testable, as maintenance will be the responsibility of the customer.

1.5.5 **Business Rules**

Users should not be allowed to add items to a cart, checkout, or view their account information unless they are logged in. Administrators should have the ability to assist users in resetting credentials. They should also be able to search for order details using a reference number for administrative purposes. Finally, administrators should be able to retrieve a list of all orders, past and present, for reporting purposes.

# 2. Test Strategy:

* Project follows Agile processes having 2-4 weeks long sprints
* Testing team will follow Hybrid testing model which includes Manual and Automated test techniques
* UI testing will be performed using Automated test frameworks. At least 60% UI test coverage via automation is required.
* API testing will also be performed using Automated test frameworks. At least 60% API test coverage via automation is required.
* Backend system testing including DB testing will be performed manually.
  1. Test levels  
     2.1.1 **Component testing**
* All of these components should be thoroughly tested to ensure that they address the use cases
* Component-level testing will also ensure that the code is maintainable, effective, robust, performant, and lacks faults.
* This is a join responsibility of Dev and QA teams
* JUnit tool can be utilized to build the solid Unit test frameworks

2.1.2 **Integration Testing**  
  
The E-Commerce System will also need to go through integration testing. Integration testing allows testers the ability to “discover faults and bugs in the interaction between integrated components”  
Once all of the components mentioned above are thoroughly tested on their own, test cases will be made to evaluate the collaboration of components.

2.1.3 **System Testing**

After integration tests are complete, the E-Commerce Website will need to go through a series of system tests, which assess whether or not the website as a whole meets functional and non-functional requirement.

For example, does the website possess all of the functionalities shown in the activity diagram?

Is the E-Commerce website fast, reliable, available, secure, and maintainable? The testers at this point need to view the system from a user perspective, and test the system in an environment that is as close to production-level as possible.

2.1.4 **Acceptance Testing**

Finally, the E-Commerce Website will go through acceptance testing.

This process will involve customers/Business Partners to ensure that the system provides them with all of the functionality they need to quickly search for and purchase items. Administrators will need to provide feedback regarding whether or not they can quickly and easily access reports, and vendors will need to determine if they can adequately use the website to sell new products, and view and update inventories. Acceptance testing will take

# **Test processes and techniques 2.2.1 Review Process** All the test artifacts including but not limited to test cases, automation test scripts etc. should go through Technical reviews.

# Technical reviews aim to ensure that test objects meet their specifications.

# While this strategy involves a review meeting, the author typically does not intend. Instead, the reviewers do a considerable amount of preparation work to review the object and provide their feedback.

# The meeting involves a recorder who takes notes, consolidates feedback from the reviewers, and generates the final document with the results

# In addition to feedback, these review meetings involve discussions around possible alternatives, as well as possible errors and defects.

# The reviewers must each have a certain level of technical expertise in order for this process to be successful.

# 2.2.2 **Test Techniques for functional testing**

* **Equivalence Class Partitioning**

**From Ecommerce product perspective -** This approach helps testers systematically generate test cases based on domains for various inputs. “An equivalence class is a set of data values that the tester assumes are processed in the same way by the test object”. In other words, this strategy involves identifying the parameters that a function accepts, generating potential valid and invalid inputs for that parameter, creating an equivalence class that represents a group of similar inputs, selecting a representative input for each equivalence class, and then generating test cases that eliminate redundancy as much as possible.

* **Boundary Value Analysis**

**From Ecommerce product perspective -** Boundary value analysis will be used in conjunction with equivalence class partitioning in order to generate test cases for the E-Commerce Website. This strategy focuses on testing inputs that lie on the boundaries between equivalence classes. “Boundary value analysis delivers a very reasonable addition to the test cases that have been identified by equivalence class partitioning. Faults often appear at the boundaries of equivalence classes. This happens because boundaries are often not defined clearly or are misunderstood”

* **State Transition Testing**

**From Ecommerce product perspective -** This testing method is beneficial in evaluating how a system reacts to transitions in state. For example, the E-Commerce Website involves a number of objects. These objects change and react to user input. For example, a user’s Cart (an object) begins in an empty state. When a user views an item and then selects “Add to Cart”, the state of the object changes—it is no longer empty. This testing strategy will be applied to see how events related to a customer searching, selecting, and purchasing an item trigger changes in the state of different objects.

* **Use-Case-Based Testing**

**From Ecommerce product perspective -** Use-case-based testing will be important in evaluating the functionality of the E-Commerce Website. This strategy involves testing the execution of paths aligned with each of the use cases illustrated in the use case diagram for the E-Commerce Website (in the design documents). For example, one of the test cases describes how a customer needs to be able to narrow down their search by applying filters. This means that at least one test case needs to be generated to evaluate whether or not the website implements that functionality correctly, and whether or not the search results are refreshed when a customer adjusts the filters.

* **Experience-Based Testing**

**From Ecommerce product perspective -** Experience-based testing, also known as intuitive-based testing, will be used to supplement black box and white box techniques in the testing of the E-Commerce Website. Intuitive-based tests leverage the skills and experience of developers and testers to generate test cases that might otherwise be overlooked in systematic testing. In this approach, “The test cases are based on where faults have occurred in the past or the tester’s ideas of where faults might occur in the future”

# Roles and responsibilities

Include the need for people and team’s capability  
This is to select the best team that can deliver the project as per the needs and schedule  
Who will do whatHere we define the roles and duties for the team, it may be by the person’s name or by the designation and title  
This helps to create clear roles and responsibilities

|  |  |
| --- | --- |
| **Role** | **Responsibilities** |
| Project Manger |  |
| Project Lead |  |
| QA |  |

# Environment requirements Hardware and software ware requirements

Following environments will be required.

* Dev Environment – Dev team owns the setup and maintenance
* Test Environment – Test team owns the setup and maintenance
* (Optional) Automation test Environment – Test team owns the setup and maintenance
* UAT Environment – Test team owns the setup and maintenance

Needs and details of the environment with configuration to be laid out  
It will also list if any specific tools or apps need to be set up for Automation

5. Test Data  
This is important for Automation Testing and Performance testing

* Source of data – DB and API’s
* Data can be generated by some automation
* Database Refresh is required for automation

# 6. Deliverables

# Deliverables can be complete and working test scripts

# The final report with pass/fail and defects

# Based on requirements this can have more or fewer items

# 7. Testing tools:

# Based on the results of Teams prior analysis and studies, following tools shall be used.

|  |  |  |
| --- | --- | --- |
| UI Automation Tool | |  | | --- | | Playwright with JavaScript, Visual Studio Code | |
| API Testing tools | SoapUI, Postman |
| Test Management Tools | Confluence, Jira, Zephyr, ALM, TestLink, Git, GitHub |
| Defect Tracking tool | Jira |
| Performance Test Tool | JMeter, Postman |
| Real-time test devices/browsers | BrowserStack Cloud |

# 8. Industry standards to follow:

This section describes the industry standard to produce high quality system that meets or exceeds customer expectations.

# 10. Testing metrics:

# This section describes the metrics that should be used in the project to analyze the project status.

## **Entry Criteria**

* The entry criteria refer to the desirable conditions in order to start test execution
* Entry criteria are flexible benchmarks. If they are not met, the test team will assess the risk, identify mitigation actions and provide a recommendation.

|  |  |  |  |
| --- | --- | --- | --- |
| **Entry Criteria** | **Test Team** | **Technical Team** | **Notes** |
| Test environment(s) is available | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png |  |  |
| Test data is available |  |  |  |
| Code has been merged successfully |  |  |  |
| Development has completed unit testing |  |  |  |
| Test scripts are completed, reviewed and approved by the Project Team |  |  |  |

## **Exit criteria**

* The exit criteria are the desirable conditions that need to be met in order proceed with the implementation.
* Exit criteria are flexible benchmarks. If they are not met, the test team will assess the risk, identify mitigation actions and provide a recommendation.

|  |  |  |  |
| --- | --- | --- | --- |
| **Exit Criteria** | **Test Team** | **Technical Team** | **Notes** |
| 100% Test Scripts executed | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png |  |  |
| 90% pass rate of Test Scripts |  |  |  |
| No open Critical and High severity defects |  |  |  |
| All remaining defects are either cancelled or documented as Change Requests for a future release |  |  |  |
| All expected and actual results are captured and documented with the test script |  |  |  |
| All test metrics collected based on reports from daily and Weekly Status reports |  |  |  |
| All defects logged in Defect Tracker/Spreadsheet |  |  |  |
| Test environment cleanup completed and a new back up of the environment |  |  |  |

# 11. Requirement Traceability Matrix:

# Requirement traceability matrix is used to trace the requirements to the tests that are needed to verify whether the requirements are fulfilled.

# 12. Risk and mitigation

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Risk** | **Mitigation Plan** | **Impact** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 13. **Reporting**

# Testing team will send weekly test progress reports to the identified stakeholders over an email

# Testing team will send formal SV handoff along with sprint completion report and all the deliverables on completion of a sprint to the identified stakeholders over an email

# Testing team will send Test Summary/Completion report on the completion of product release

* Team shall leverage Jira Zephyr test reports for manual execution
* Team shall leverage Playwright/Allure test reports for Automated test execution

# 15. Approvals:

The following people are required to approve the Test Strategy

|  |  |
| --- | --- |
| **Approved By Role** | **Approved By Name** |
|  |  |
|  |  |
|  |  |