Test Plan

VERSION 1.0

Preparate by:

Yordan Nikolov

Table of Contents

IV. Testing Strategy

1. Testing types
    - Smoke testing
    - Functional testing

2. Testing techniques
    - Boundary Value Analysis
    - Equivalence partitioning
    - Classification trees

3. Entry Criteria

4. Exit Criteria

V. Environment

VI. Responsibilities

VII. Project Timeline

- Risks
- Mitigation Plan

I. Introductions

Spree is a headless open-source ecommerce platform for multi-language and multi-currency global brands. As we explore the intricacies of this e-commerce solution, we'll navigate through its headless architecture, delve into its multi-language and multi-currency support, and examine the functionalities that form the backbone of a seamless online shopping experience. The demo site serves as a playground for testing and exploring the platform's capabilities in real-time. The application provides different functionalities tailored to various user roles:

User Management:

- Registration, login, and account management functionalities.
- Password recovery/reset mechanisms.

Product Management:

- Adding, editing, and deleting products.
- Product categorization and filtering.

Shopping Cart:

- Adding/removing items from the shopping cart.
- Total amount calculation.

Checkout Process:

- Various steps in the checkout process.
- Payment method verification.

Localization:

- Language switching functionality.
- Currency display and conversion.

Search Functionality:

- Search different products.

Product CRUD Operations:

- Capability to add, edit, and delete products, checking for accuracy and ease of use.

II. Objectives and Tasks

Objectives:

The primary objective of testing is to ensure that the system fully meets its requirements, both functional and non-functional. The aim is to validate that the quality metrics for each requirement are satisfied, the use case scenarios are well-supported, and the overall product quality is maintained. The secondary objectives involve identifying and exposing all issues and associated risks.

Tasks:

The following tasks are to be completed by the entire QA team:

Planning and control:

- Conduct a detailed analysis of the Spree website project Specifications and requirements.
- Distribute roles and responsibilities.
- Define the scope of testing, including modules and features to be tested.
- Set entry and exit criteria for the entire project.

Analysis and design:

- Prioritize features and functionalities for testing based on their importance and impact.
- Select appropriate testing techniques and methodologies.
- Estimate the timeframe required for testing individual features.
- Allocate time estimates for different testing phases, such as exploratory testing, manual testing, and automated testing.

Implementation and execution:

- Develop detailed test cases based on the prioritized features.
- Conduct peer reviews of test cases and execute them.
- Identify test cases suitable for automation.
- Develop automated tests for both API and UI components
- Execute automated tests and review results.
- Prepare comprehensive test reports and bug reports.

III. Scope

The primary focus is to test the essential functionalities as outlined in the DEV team's document. Additional features developed by the DEV team will be tested as time permits.

1. Features To Be Tested:

- ***Registration form;***
- ***Login form;***
- ***Account management functionalities;***
- ***Password recovery***
- ***Adding, editing, and deleting products.***
- ***Product categorization and filtering.***
- ***Adding/removing items from the shopping cart.***
- ***Total amount calculation.***
- ***Various steps in the checkout process.***
- ***Payment method verification.***
- ***Language switching functionality.***
- ***Currency display and conversion.***

2. Features NOT To Be Tested:

- **Social Media Icons:**

Reason for Not Testing: Social media icons are considered external links and are typically static. As such, they are excluded from functional testing unless there are specific interactions or functionalities associated with them.

- **Filter Options:**

Manufacturer and Brand Filters:
Reason for Not Testing: Manufacturer and brand filters are assumed to be standard filtering features and are not explicitly tested unless there are indications of issues or specific requirements.

- **Promo Code:**

Reason for Not Testing: The functionality related to applying promo codes is often considered stable and is only tested if there are reported issues or changes in this feature.

- **Payment Method - Credit Card:**

Reason for Not Testing: Credit card payment processing is typically a well-established and stable third-party service. Direct testing of credit card transactions is usually excluded from functional testing due to security and compliance considerations.

- **Billing Address Different Than User Address:**

Reason for Not Testing: This feature is commonly supported by e-commerce platforms and is only tested if there are specific requirements or reported issues.

- **Delivery Methods:**

Reason for Not Testing: Delivery methods are often part of standard checkout functionality and are only tested if there are changes or specific requirements.

- **Main Page Sections:**

Bestsellers, Fashion Trends, Trending, Street Style:
Reason for Not Testing: Content-based sections on the main page, such as trends and promotions, are often dynamic and subject to change based on business decisions. Testing these sections is typically reserved for regression testing or when there are specific updates.

IV. Testing Strategy

1. Testing types

***Smoke testing*:**

Also known as build verification testing, smoke testing focuses on the main functionalities of the software to ensure they work as intended before more detailed testing begins. This phase will involve both exploratory and manual test on key features. The goal is to quickly identify severe failures that could halt further testing.

***Functional testing*:**

Functional testing aims to validate that the software behaves according to its specified requirements. During manual testing, decisions will be made regarding which test cases to automate. Automated tests will be used for regression and smoke tests to ensure that no regression issues affect key functionalities. This allows for rapid feedback on the e-commerce status after each test cycle.

2. Testing techniques:

Boundary Value Analysis:

This technique focuses on testing the boundary value of valid and invalid partitions. It aims to identify defects that are more likely to occur at the edges of these partitions. Boundary Value Analysis tests input values close to the boundaries, as they have a higher likelihood of errors.

Equivalence partitioning:

In this technique, input data is divided into partitions of valid and invalid values. All partitions should exhibit the same behavior. Test cases are designed to cover each partition at least once, ensuring that if one condition in a partition is true, the same should hold for other equivalent partitions.

Classification trees:

Classification trees are used to create a set of if-then logical conditions for accurate prediction or classification of cases. This technique is particularly useful when the dataset needs to be divided into classes corresponding to the response variable, often binary classes like Yes or No.

3. Entry Criteria:

Entry criteria serve as the prerequisites that must be met before initiating a specific task. The following conditions must be satisfied to officially commence testing:

- The software has been delivered.
- Test data for various test cases is prepared and available.
- Requirements have been clearly outlined and approved.
- High-level test cases have been developed and are prepared for execution.
- The test environment is configured, and all essential resources, including tools and devices, are in place and available to use by the QA team.

4. Exit Criteria:

Exit criteria are the conditions that must be met to officially conclude a specific task. Meeting these criteria signifies the completion of the QA team's work:

- All high-priority functionalities have a 100% pass rate for both automated and manual test cases.
- At least 50% of test cases for medium-priority functionalities pass successfully in both automated and manual testing.
- The allocated time for testing has been exhausted.

Priority Scale and definition

Priority Scale and definition		
1	**Highest**	They are blockers. Critical service issues affecting all end users. Service unavailable or unusable with no workaround.
2	**High**	The service issue is critically impacting a significant proportion of end users or critically impacting collaboration among end users.
3	**Medium**	The service issue affects an individual, a small number of employees, a non-critical function or a workaround is easily available.
4	**Low / Lowest**	The service issue affects an individual or small group of Customer who requests a change to the functional specification. Product improvements that will not affect the software usage in any way and can be postponed, if there are tight deadlines.

Severity Scale and definition

Severity Scale and definition		
1	**Blocking**	Users are not able to perform a task or complete a scenario with the design as implemented. Important features are missing, broken, or behaving in a way that users won't be able to understand or remedy. Reasonable user action results in data loss. Spelling mistakes in main functionalities or names.
2	**High**	This issue may lead a customer to unknowingly put their site or system into an undesirable state. Any issue that erroneously reports system status. Significant interaction, layout, or other visual defects that most users will notice, and will slow or block users from completing important tasks. Clear negative impact to the perception of quality.
3	**Medium**	Minor loss of function or other issue where an easy workaround exists. Moderate interaction, layout, or visual defects that do not block or slow a user from completing a desired task, but are noticeable.
4	**Low**	Minor usability and formatting issues. Layout or visual defects that are likely to go unnoticed by users.

V. Environment

Given the absence of specific requirements for the test environment, the team has opted to conduct all testing under the following conditions:

- Web Browser: Latest version of Google Chrome (currently 117.0.5938.132 Official Build, 64-bit).
- Operating System: Windows 10, 64-bit.
- Processor: Intel® Core™ i7-6800K.
- RAM: 32GB.
- OS Details: Windows 10 Home, Version 22H2 (OS Build 19045.3448).

VI. Responsibilities:

Creation of Jira Xray Project	Yordan Nikolov
Creation of GitHub Repository	Yordan Nikolov
Creation of Test Case template	Yordan Nikolov
Creation of Bug Report template	Yordan Nikolov
Test Automation Framework	Yordan Nikolov

VII. Resources:

Communication Channels:

- Microsoft Teams
- Microsoft OneNote

Version Control:

- GitHub

Task Management Board:

- Jira

Manual Tools:

- Jira
- Microsoft Excel

Automation Tools:

- **Postman v10.18.9**
- **IntelliJ IDEA 2023.2 (Ultimate Edition)**
- **Selenium WebDriver**

VIII. Project Timeline:

The total duration allocated for completing all activities:

Task	Date
Requirement documents review	01.12 – 03.12
Test plan and documentation	04.12 – 06.12
Initial Set Up	06.12 – 08.12
Exploratory Testin	08.12 – 09.12
Manual Test Cases	09.12 – 14.12
API Tests Automation	15.12 – 21.12
Re-Testing	21.12 – 24.12
Test Reports	26.12 – 28.12
UI Tests Automation	29.12 – 06.01

Risk:

Strategies to mitigate these risks:

- Tight Testing Schedule and Incorrect Time Estimates: Underestimation of the time required for testing activities.
- Unexpected Unavailability of QA Member: Sudden absence or unavailability of key team member.
- Unexpected Complexities in Feature Testing: Encountering unanticipated challenges while testing specific features.

Mitigation Plan:

Strategies to mitigate these risks:

- Regular Daily Communications: Maintain constant communication among team members to quickly identify and address issues.
- Overtime Work: The test team is prepared to work extended hours to meet deadlines.
- Scope Adjustment: The scope of the test plan may be modified to accommodate unforeseen challenges.