

Software Test Plan

petstore



Mahmood Odeh

contents

contents	2
1. Introduction	4
a. Purpose	4
b. Project Overview	4
c. Website Overview.....	4
d. Key Features.....	4
2. Test Strategy	5
a. Test Objectives	5
b. Test Assumption.....	5
c. Test Approach.....	7
d. Data Approach.....	7
e. Levels of Testing	9
3. Execution Strategy	13
a. Test environment setup	13
b. Test Execution	13
c. Test Automation	13
d. Defect Management.....	13
e. Reporting and Monitoring	14
f. Continuous Improvement	14
g. Defect tracking & Reporting	14
4. TEST MANAGEMENT PROCESS.....	14
a. Test Planning	15
b. Test Design	15
c. Test Execution.....	15
e. Defect Management	15
f. Test Closure.....	15
g. Continuous Improvement	16
5. Test Environment	16
6. Approvals.....	16

1. Introduction

a. Purpose

The purpose of this document is to:

- Define the scope of testing for the PetStore.com website.
- Describe the testing objectives and approach.
- Identify the resources, roles, and responsibilities for testing.
- Outline the test environment and tools.
- Provide a schedule for testing activities.
- Specify the deliverables and criteria for test completion.

b. Project Overview

The Software Test Plan (STP) document outlines the testing approach and strategy for ensuring the quality of the PetStore.com website. This document will guide the testing team in planning, executing, and reporting on the testing activities.

c. Website Overview

PetStore.com is an online platform that offers a wide range of pet products for sale. The website allows users to browse and purchase pet supplies, including food, toys, accessories, and more. Users can create accounts, add items to their cart, and complete purchases through the website.

d. Key Features

- User Registration and Authentication: Users can create accounts, log in, and manage their profiles.
- Product Browsing: Users can browse products by category, search for specific items, and view product details.
- Shopping Cart: Users can add products to their cart, view the cart contents, and proceed to checkout.
- Checkout and Payment: Users can complete purchases using various payment methods.
- Order Management: Users can view their order history and track the status of their orders.
- Admin Dashboard: Admins have access to a dashboard for managing products, orders, and users.

- Responsive Design: The website is optimized for use on desktop and mobile devices.

2. Test Strategy

a. Test Objectives

The primary objective of the testing strategy for PetStore is to ensure that the website functions as intended, meets the specified requirements, and provides a positive user experience. The testing will cover functional, non-functional, and usability aspects of the website.

The final product of the test is:

- A working product with minimal bugs that meets the specifications.
- A set of stable test scripts that can be reused for test execution and user acceptance tests.

b. Test Assumption

key assumption:

- Production like data required and be available in the system prior to start of Functional Testing

General assumptions:

- Availability of Test Environments: It is assumed that a dedicated test environment, mimicking the production environment, is available for testing purposes without interference from ongoing development activities.
- Tool Availability: Necessary testing tools are available and configured for use by the testing team.
- Documentation: Comprehensive documentation on application features, known issues, and integration points is available for reference by the test team.

- Dev team will provide Defect fix plans based on the Defect meetings during each cycle to plan.
- Schedule Adherence: All parties involved will adhere to the testing schedule, including timely completion of development sprints, bug fixes, and re-testing phases.
- The defects will be tracked through Jira only. Any defect fixes planned will be shared with Test Team prior to applying the fixes on the Test environment
- The system will be treated as a black box; if the information shows correctly online and in the reports, it will be assumed that the database is working properly.

Functional and Non-Functional Testing:

- The testing team will perform functional and non functional tests at the most of the website.

User Acceptance Testing:

- UAT test execution will be performed by independent users and QA Group will provide their support on creating UAT script.

c. Test Approach

- Manual Testing: Manual testing plays a crucial role in assessing the application's user interface and user experience. It involves testers interacting with the application, executing test cases, and observing outcomes to identify any discrepancies from expected behavior. This approach is particularly effective in evaluating the usability and aesthetic aspects of the application, such as layout, design consistency, and intuitive navigation.
- Automated Testing: Automated testing complements manual efforts by executing repetitive and regression test

cases more quickly and efficiently. Using automated testing tools, scripts were developed to simulate user actions on the Login Page, Sign Up Page, Home Page, Profile Page. This approach enables the rapid identification of defects and supports continuous integration and delivery processes.

d. Data Approach

1. Test Data Requirements

- User Data: Create test user accounts with different roles (guest, registered user, admin) to test various functionalities such as registration, login, and account management.
- productData: Generate test product listings with different attributes (category, price, availability) to test browsing, searching, and purchasing functionalities.
- order Data: Create test orders with different products, quantities, and payment methods to test the checkout process.

2. Test Data Management

- Data Storage: Store test data in a database or file system, ensuring it is easily accessible and can be updated as needed.
- Data Versioning: Version test data to track changes and ensure consistency across test runs.
- Data Masking: Mask sensitive data (e.g., personal information, payment details) to ensure privacy and security during testing.

3. Test Data Generation

- Automated Data Generation: Use automated scripts to generate test data, especially for large datasets or complex scenarios.
- Data Variation: Generate test data that covers a wide range of values and conditions to thoroughly test the website's functionality.

4. Test Data Reusability

- **Data Templates:** Create reusable data templates for common test scenarios (e.g., adding a product to the cart, updating user information).
- **Data Dependency Management:** Manage dependencies between test data to ensure tests can be executed in isolation.

5. Test Data Quality

- **Data Validation:** Validate test data to ensure it meets the required format and constraints (e.g., valid email addresses, correct currency format).
- **Data Cleansing:** Cleanse test data to remove duplicates, inconsistencies, or irrelevant information.

6. Test Data Security

- **Data Encryption:** Encrypt sensitive test data to ensure it is protected during storage and transmission.
- **Access Control:** Implement access controls to restrict access to test data based on roles and permissions.

e. Levels of Testing

■ Exploratory

PURPOSE: the purpose of this test is to make sure critical defects are removed before the next levels of testing can start.

TESTERS: Testing team.

METHOD: this exploratory testing is carried out in the application without any test scripts and documentation

Timing: At the start of the testing phase.

■ Functional and non Functional Tests

Functional Test

Purpose: The primary purpose of functional testing is to test each function of the software application, by providing appropriate input, and verifying the output against the Functional Requirements.

TESTERS: Testing team.

METHOD: The test will be performed according to the Functional test cases that are stored in our Testrail environment.

Timing: After the Exploratory Testing has been complete.

Non Functional Test

Purpose: The primary purpose of non-functional testing is to evaluate the non-functional aspects of a software application, which include performance, usability, reliability, security and localization.

TESTERS: Testing team.

METHOD: The test will be performed according to the Non-Functional test cases that are stored in our Testrail environment.

Timing: After the Functional Testing has been complete

Test Tree

1. UI Tests

- a. **Homepage**
 - i. **Verify the presence of key elements (e.g., logo, navigation menu, search bar).**
 - ii. **Test navigation links.**
- b. **Product Listing Page**
 - i. **Verify products are displayed correctly.**
 - ii. **Test sorting and filtering options.**
- c. **Product Details Page**
 - i. **Verify product details are displayed accurately.**
 - ii. **Test the "Add to Cart" functionality.**
- d. **Cart Page**
 - i. **Verify the contents of the cart.**

- ii. Test the checkout process.
 - e. Checkout Process
 - i. Test the different steps of the checkout process (e.g., shipping, payment).
 - f. User Account
 - i. Test user registration and login.
 - ii. Test profile management functionalities.
- 2. Functional Tests
 - a. User Management
 - i. Test user registration, login, and logout.
 - ii. Test user profile management.
 - b. Product Management
 - i. Test adding, editing, and deleting products.
 - c. Order Management
 - i. Test placing orders and order status updates.
- 3. Non-Functional Tests
 - a. Performance Testing
 - i. Test website response times under various loads.
 - ii. Test website scalability.
 - b. Compatibility Testing
 - i. Test website functionality on different browsers (Chrome, Firefox, Safari, Edge).
 - c. Security Testing
 - i. Test for vulnerabilities such as XSS, CSRF, SQL injection.
 - ii. Test data encryption and secure communication.
 - d. Usability Testing
 - i. Test website navigation and user interface.
 - ii. Gather feedback from users for improvements.
- 4. API Tests
 - a. Authentication
 - i. Test API authentication mechanisms.
 - b. Product API
 - i. Test endpoints for adding, editing, and deleting products.
 - c. Order API
 - i. Test endpoints for placing orders and updating order status.
- 5. End-to-End Tests
 - a. Test the entire user journey from browsing products to completing a purchase.
 - b. Test different user scenarios (e.g., new user, returning user, admin user).

■ User Acceptance Test

PURPOSE: end users or clients test the software to ensure it can handle required tasks in real-world scenarios, according to specifications.

TESTERS: Independent outside users .

METHOD: End users execute the test cases, performing tasks as they would in their day-to-day work. This includes navigating through the application, inputting data, and using the features and functions of the system.

Timing: After the Functional and Non-Functional tests

3. Execution Strategy

a. Test environment setup

Environment Configuration: Ensure the test environment is configured to match the production environment as closely as possible. Data Setup: Prepare test data, including user accounts, accommodations, orders, and other relevant data for testing.

b. Test Execution

Unit Testing: Developers will perform unit tests using JUnit/TestNG for backend code and Selenium for frontend code. Integration Testing: Test integration between different modules and components to ensure they work together correctly.

System Testing: Test the entire system to verify that all features and functionalities work as expected.

Acceptance Testing: Conduct acceptance tests with stakeholders to validate that the system meets the requirements.

c. Test Automation

Regression Testing: Automate regression tests to ensure that new changes do not break existing functionality. Smoke Testing: Automate smoke tests to quickly verify that the critical functionalities are working after each build.

d. Defect Management

- Defect Identification: Use tools like Jira to track and manage defects.
- Defect Prioritization: Prioritize defects based on severity and impact on the system.
- Defect Resolution: Developers will resolve defects, and testers will verify the fixes.

e. Reporting and Monitoring

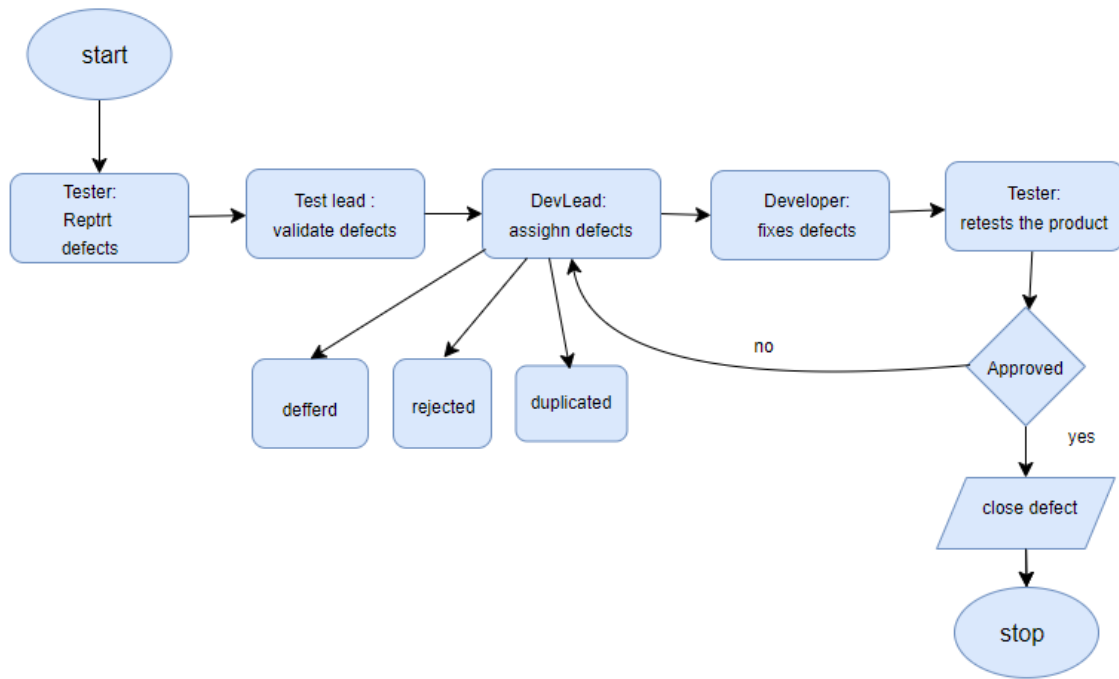
Test Execution Reports: Generate and share test execution reports to track progress and identify issues. Monitoring: Monitor the test execution process to ensure that it is on track and meets the timeline.

f. Continuous Improvement

- Feedback Loop: Collect feedback from stakeholders and team members to identify areas for improvement.
- Process Optimization: Continuously optimize the testing process to improve efficiency and effectiveness.

g. Defect tracking & Reporting

- The presented flowchart illustrates the Defect Tracking Process:



4. TEST MANAGEMENT PROCESS

a. Test Planning

- Define Test Strategy: Define the overall testing strategy, including the use of Selenium Grid for parallel test execution.
- Identify Test Scenarios: Identify test scenarios for each layer of the application (infrastructure, logic, UI).
- Create Test Plan: Create a detailed test plan outlining the testing approach, resources, schedule, and deliverables.

b. Test Design

- Create Test Cases: Develop test cases for each test scenario, specifying the steps to be executed and the expected results.
- Implement Test Logic: Implement the test logic for each test case, including setting up the test environment and configuring Selenium Grid for parallel execution.

c. Test Execution

- **Execute Tests:** Run the tests using PyCharm and Selenium, utilizing Selenium Grid, Jenkins to run tests in parallel across different browsers and environments.
- **Monitor Execution:** Monitor the test execution process to ensure tests are running correctly and any issues are promptly addressed.

d. Test Reporting

- **Generate Reports:** Generate test reports using PyCharm or other reporting tools like Jenkins to track test execution results and identify any failures.
- **Analyze Results:** Analyze test results to identify trends, patterns, and areas for improvement.

e. Defect Management

- **Identify Defects:** Use PyCharm or a defect tracking tool to identify and log defects found during testing.
- **Prioritize Defects:** Prioritize defects based on severity and impact on the application.
- **Resolve Defects:** Work with the development team to resolve defects, retesting as necessary to verify fixes.

f. Test Closure

- **Evaluate Testing:** Evaluate the testing process to identify successes and areas for improvement.
- **Document Lessons Learned:** Document lessons learned from the testing process for future projects.
- **Prepare Test Closure Report:** Prepare a test closure report summarizing the testing activities, results, and any outstanding issues.

g. Continuous Improvement

- **Review and Update:** Regularly review and update the test plan, test cases, and test strategy based on feedback and lessons learned.
- **Training and Development:** Provide training and development opportunities for team members to improve their testing skills and knowledge.

5. Test Environment

Platforms:

- pet store website
- Windows 11.

Devices:

- A range of desktop computers devices have the firefox,chrome and edge

Network Conditions:

- Tests were performed under various network speeds and conditions (Wi-Fi, 4G, 3G) to assess performance and reliability.

Test Tools:

- pycharm
- selenium
- selenium grid
- jenkins

6. Approvals

Name	Role	Signature
Mahmood Odeh	Project Management	
Tsahe Agindar	Test Lead	
yair amon	Business Analyst	