### **1. Test Plan Identifier**

* **TP-PETSTORE-01**

### **2. References**

* PetStore Functional Specification Document
* PetStore User details
* PetStore System Architecture Document
* PetStore Design Documents
* PetStore Database
* Test Strategy Document

### **3. Introduction**

The purpose of this test plan is to outline the testing strategy for the PetStore application. The PetStore is an online e-commerce platform that allows customers to purchase pets. This plan defines the testing objectives, approach, resources, and schedule required to verify and validate the functionality, performance, and reliability of the PetStore system.

### **4. Test Items**

* **PetStore Web Application** (Web interface for customers)
* **PetStore Admin Panel** (Web interface for managing products and orders)
* **Database** (Backend database that stores user data, orders, and inventory)

### **5. Software Risk Issues**

* **Data Integrity**: Potential risks related to incorrect data handling, especially in inventory management and user orders.
* **Payment Gateway Integration**: Risks related to third-party payment gateways (e.g., credit card processing failures, security vulnerabilities).
* **High Traffic Load**: The website may experience a high volume of users, which could affect performance.
* **Security**: Ensuring the application is secure, especially user data (e.g., passwords, payment info).
* **Cross-Browser Compatibility**: Risk of the application not working consistently across different browsers.

### **6. Features to be Tested**

* **User Registration and Login**: Ensuring the user can sign up, login, and log out successfully.
* **Product Search and Browsing**: Verifying that customers can search for pets and pet supplies, and view product details.
* **Shopping Cart**: Testing add/remove items from the shopping cart, quantity updates, and price calculations.
* **Checkout Process**: Validating the purchase flow, including payment gateway interaction, order confirmation, and shipping details.
* **Admin Panel**: Testing inventory management, order tracking, and customer management.
* **Security**: Testing user authentication, encryption, and secure payment processing.

### **7. Features not to be Tested**

* **Third-Party API Integration**: Testing of third-party services like payment gateways (since these are handled externally).
* **Legacy Code**: Parts of the system that are not part of the current version will not be tested (unless modified).

### **8. Approach**

* **Manual Testing**: Functional tests, UI testing, and cross-browser testing.
* **Automated Testing**: Unit tests for backend services, API tests, and regression tests using Selenium for front-end.
* **Performance Testing**: Load testing using tools like JMeter to simulate user traffic.
* **Security Testing**: Penetration testing and vulnerability scanning using OWASP tools.
* **Exploratory Testing**: Manual exploratory testing by testers to identify edge cases or potential issues.

### **9. Item Pass/Fail Criteria**

* **Pass**: The feature meets the expected functionality, performance, and security criteria, and does not produce any critical bugs.
* **Fail**: The feature does not meet the expected behavior or generates defects that prevent users from completing key tasks (e.g., purchase).

### **10. Suspension Criteria and Resumption Requirements**

* **Suspension Criteria**: Testing will be suspended if critical defects are identified, such as inability to log in, checkout failure, or system crashes.
* **Resumption Requirements**: Testing will resume after the critical defect has been resolved and the fix has been verified.

### **11. Test Deliverables**

* **Test Plan Document** (This document)
* **Test Cases** (Detailed test cases for all features)
* **Test Logs** (Execution logs of test runs)
* **Defect Reports** (Details of all defects identified)
* **Test Summary Report** (A report summarizing test execution, coverage, and defect status)

### **12. Remaining Test Tasks**

* Finalize and review test cases.
* Execute regression tests on new builds.
* Perform exploratory testing on the admin panel and user flow.
* Conduct security testing.

### **13. Environmental Needs**

* **Test Environment**: A dedicated test server, with the same configuration as production, including a staging database.
* **Tools**: Selenium for automated UI tests, JUnit for backend unit tests, JMeter for load testing.
* **Browsers**: Chrome, Firefox, Safari, Edge for cross-browser testing.

### **14. Staffing and Training Needs**

* **Testers**: 2 QA engineers for functional testing, 1 QA engineer for performance testing, and 1 security tester.
* **Training**: Testers need to be trained in using testing tools (e.g., Selenium, JMeter) and familiar with the PetStore application.
* **Development Team Support**: Available for fixing defects and providing technical assistance.

### **15. Responsibilities**

* **Test Lead**: Oversee the testing process, create test plan, manage the testing team, and report progress.
* **QA Engineers**: Execute functional, performance, and security tests, log defects, and provide feedback.
* **Development Team**: Address defects found during testing and provide clarifications as needed.
* **Project Manager**: Ensure that testing is completed according to schedule and report status to stakeholders.

### **16. Schedule**

* **Test Planning**: 1 week (October 16 - October 23)
* **Test Case Design**: 1 week (October 24 - October 30)
* **Test Execution**: 3 weeks (November 1 - November 21)
* **Bug Fixing and Retesting**: 2 weeks (November 22 - December 5)
* **Final Reporting**: 1 week (December 6 - December 12)

### **17. Planning Risks and Contingencies**

* **Risk**: Delays in defect resolution could push back the testing schedule.
  + **Contingency**: Prioritize high-severity defects and allocate more resources if necessary.
* **Risk**: Unavailability of key team members (testers, developers).
  + **Contingency**: Cross-train team members to mitigate this risk.

### **18. Approvals**

* **Test Plan Approved By**:
  + [Test Lead Name]
  + [Project Manager Name]
  + [Product Owner Name]

### **19. Glossary**

* **UI**: User Interface
* **API**: Application Programming Interface
* **JMeter**: Performance testing tool
* **Selenium**: Automation tool for web applications
* **Regression Testing**: Re-testing of features after code changes
* **Penetration Testing**: Security testing to identify vulnerabilities