Assignment- E-Commerce Website Test Plan

Pradipta Chatterjee

QA

10/12/2023

Test Strategy Document for E-commerce Website

1. Testing Objectives:

1.1 Functional Testing Objectives

- Ensure all features and functionalities of the e-commerce website work as intended.
- Verify a seamless user experience from product browsing to checkout.
- Validate proper integration with external systems such as payment gateways and shipping providers.

1.2 Performance Testing Objectives

- Assess the website's response time, scalability, and resource utilization.
- Identify and address performance bottlenecks to ensure optimal user experience.
- Evaluate the website's ability to handle a specified number of concurrent users.

1.3 Security Testing Objectives

- Verify the confidentiality and integrity of user data.
- Ensure secure handling of payment transactions.
- Identify and mitigate vulnerabilities such as SQL injection, Cross-Site Scripting (XSS), and other security threats

2. Scope of Testing: This test strategy covers functional, performance, and security testing of the e-commerce website http://live.guru99.com/. It also includes integration testing with third-party services, browsers, and devices.

3. Testing Levels:

3.1 Unit Testing

Developers will conduct unit testing to verify the functionality of individual components and modules.

3.2 Integration Testing

Testers will perform integration testing to validate the interactions between different modules, services, and third-party integrations.

3.3 System Testing

Conduct comprehensive system testing to ensure the entire e-commerce application functions cohesively.

3.4 User Acceptance Testing (UAT)

Involve stakeholders in UAT to ensure the website meets business requirements and user expectations.

4. Testing Types:

4.1 Functional Testing

- Unit Testing
- Integration Testing
- System Testing
- User Acceptance Testing (UAT)

4.2 Performance Testing

- Load Testing
- Stress Testing
- Scalability Testing
- Performance Monitoring

4.3 Security Testing

- Vulnerability Scanning
- Penetration Testing
- Security Audits
- Compliance Testing

5. Entry and Exit criteria:

5.1 Entry criteria:

1. **Requirements Availability:**
    - All relevant and necessary requirements documents, including functional specifications and design documents, should be complete and available.
2. **Environment Setup:**
    - The testing environment, including hardware, software, databases, and network configurations, should be set up and ready for use.
3. **Test Resources Ready:**
    - All required testing resources such as test data, test scripts, test cases, and testing tools should be prepared and available.
4. **Build Deployment:**
    - The application builds to be tested should be deployed and accessible in the testing environment.
5. **Test Team Availability:**
    - The testing team members should be available and adequately trained on the testing processes and tools.

6. **Test Schedule Defined:**
    - A testing schedule with milestones and timelines should be defined and agreed upon.
7. **Entry Meeting:**
    - A meeting involving relevant stakeholders, including developers, testers, and project managers, should be conducted to ensure everyone is aligned on the testing goals and scope.

5.2 Exit Criteria:

1. **Test Execution Completion:**
    - All planned test cases should be executed, and the test execution phase should be completed.
2. **Defect Resolution:**
    - All critical and high-severity defects should be resolved, and their fixes should be retested and verified.
3. **Test Summary Report:**
    - A test summary report should be prepared, including details such as test coverage, test results, and any outstanding issues.
4. **User Acceptance Testing (UAT) Completed:**
    - If there is a UAT phase, it should be completed, and user acceptance criteria should be met.
5. **Performance Metrics:**
    - If performance testing is included, performance metrics should meet the predefined criteria.
6. **Regression Testing Completed:**
    - Regression testing should be performed on the modified or new features to ensure that existing functionalities are not adversely affected.
7. **Documentation Updated:**
    - All relevant documentation, including test cases, test scripts, and user manuals, should be updated to reflect the latest changes.
8. **Exit Meeting:**
    - A meeting involving relevant stakeholders should be conducted to review the testing process, discuss any issues, and obtain approval to move forward with the release.

6. Test Environment and Tools:

Recommended Operating Systems

- **Windows:** 7 or newer
- **MAC:** OS X v10.7 or higher
- **Linux:** Ubuntu

Hardware Requirements

- Processor: Minimum 2.5 GHz; Recommended 3GHz or more
- Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)
- Hard Drive(SSD): Minimum 128 GB; Recommended 512 GB or more
- Memory (RAM): Minimum 4 GB; Recommended 8 GB or above
- Sound card w/speakers

Recommended Software

Supported Browsers

- Firefox
- Chrome

Other important software

- Java
- Adobe Flash Player
- Adobe Reader

7.Risk Analysis

7.1 Security Risks:

- **Payment Security:** Ensure that payment transactions are secure, and sensitive information such as credit card details is encrypted.
- **Data Protection:** Assess how user data is handled and stored, ensuring compliance with data protection laws (e.g., GDPR).
- **Session Management:** Check for vulnerabilities related to user sessions, such as session hijacking or fixation.

7.2 Functionality Risks:

- **Transaction Flow:** Verify that the end-to-end transaction flow, from product selection to checkout, works smoothly without errors.
- **Compatibility:** Test the website on various devices, browsers, and operating systems to ensure compatibility.
- **Performance:** Evaluate website performance under different traffic loads to identify potential bottlenecks.

7.3 Usability Risks:

- **User Interface (UI):** Ensure the website has an intuitive and user-friendly interface, with easy navigation and clear calls to action.

- **Accessibility:** Confirm that the website is accessible to users with disabilities, following accessibility standards such as WCAG.

7.4 Load and Performance Risks:

- **Load Testing:** Assess the website's response time and performance under different levels of user load.
- **Scalability:** Test the website's ability to scale with increased traffic, ensuring it can handle peak loads.

7.5 Content Management Risks:

- **Product Information:** Verify that product information (e.g., images, prices, descriptions) is accurate and up-to-date.
- **Content Publishing:** Ensure that changes to website content, such as product updates or promotional banners, are reflected correctly.

7.6 Integration Risks:

- **Third-Party Integrations:** Test any third-party integrations (payment gateways, shipping providers, etc.) to ensure seamless functionality.
- **API Security:** If the website uses APIs, assess the security of these interfaces to prevent unauthorized access.

7.7 Recovery and Backup Risks:

- **Backup and Restore:** Evaluate the website's backup and restore procedures to ensure data integrity in case of system failures.
- **Disaster Recovery:** Plan for potential disasters, such as server crashes or data breaches, and assess the website's recovery capabilities.

7.8 Compliance Risks:

- **Legal Compliance:** Ensure the website complies with relevant laws and regulations, including consumer protection and e-commerce regulations.
- **Privacy Policy:** Verify that the website has a clear and comprehensive privacy policy.

7.9 User Input Validation Risks:

- **Form Validation:** Check for proper validation of user inputs to prevent common security vulnerabilities like SQL injection or cross-site scripting.

7.10 Regression Risks:

- **Regression Testing:** Perform regression testing to ensure that new updates or changes do not negatively impact existing features and functionality.

2. Test Planning:

2.1 Test Deliverables

- Test plan
- Test cases
- Defect reports
- Test summary report

2.2 Test Schedule

Test Phases

- Phase 1: Planning (15/12/2023 – 19/12/2023**)**
    - Define test objectives and deliverables.
    - Identify test resources and establish the testing environment.
- Phase 2: Test Case Design and Preparation (20/12/2023 – 27/12/2023)
    - Develop test cases for functional, security, and performance testing.
    - Prepare test data and environment.
- Phase 3: Test Execution (28/12/2023 – 15/01/2024)
    - Execute test cases.
    - Monitor and report defects.
- Phase 4: Defect Resolution and Retesting (16/01/2024 – 31/01/2024)
    - Developers address reported defects.
    - Retest to confirm defect resolution.
- Phase 5: User Acceptance Testing (01/02/2024 – 09/02/2024)
    - Involve users to ensure the system meets business requirements.
- Phase 6: Test Closure (12/02/2024 – 16/02/2024)
    - Summarize testing activities.
    - Prepare test summary report.

Test Activity

- Define entry and exit criteria for each phase.
- Identify dependencies between testing activities.
- Regularly update the testing schedule based on progress.

2.3 Test Resources

Human Resources

- Test Manager
- Testers (Functional, Security, Performance, UAT)
- Developers
- Users for UAT

Infrastructure

- Test environments (development, testing, production)
- Various devices and browsers for compatibility testing
- Security testing tools

- Performance testing tools

Test Data

- Sample user accounts for testing.
- Various product data for testing different scenarios.
- Dummy payment information for payment processing testing.

2.4 Test Execution

Test Environment Readiness:

- Ensure that the test environment is set up and stable.
- Verify that all necessary software, hardware, and network configurations are in place.
- Confirm that test data is available and accurate.

Test Data Preparation:

- Validate that test data is relevant and covers a wide range of scenarios.
- Ensure that data is sanitized and does not contain sensitive information.
- Verify that test data is properly loaded into the test environment.

Test Case Execution:

- Execute test cases based on the test plan.
- Follow the test execution schedule and prioritize test cases.
- Record actual results, including any deviations from expected results.

Test Automation:

- If applicable, execute automated test scripts.
- Validate the accuracy and reliability of automated test results.
- Address any issues related to test automation.

Integration Testing:

- Confirm that integrated components/systems interact as expected.
- Verify that data is passed correctly between integrated modules.
- Execute integration test cases to validate end-to-end functionality.

Performance Testing:

- If performance testing is part of the scope, execute performance tests.
- Monitor system resource usage and response times.
- Identify and address any performance bottlenecks.

Security Testing:

- If security testing is part of the scope, execute security tests.

- Validate the system against security vulnerabilities and threats.
- Address and document any security issues identified.

User Acceptance Testing (UAT):

- If UAT is part of the testing process, coordinate with users.
- Execute UAT test cases and record feedback.
- Ensure that UAT criteria are met before proceeding to the next phase.

Regression Testing:

- Conduct regression testing to ensure that new changes do not adversely affect existing functionality.
- Verify that previously identified issues have been resolved.

Defect Management:

- Log defects for any issues identified during testing.
- Categorize and prioritize defects based on severity and impact.
- Verify the resolution of defects and re-test as needed.

Test Reporting:

Test Summary Report:

- Provide a high-level overview of the test execution.
- Include the status of testing phases, overall test coverage, and key metrics.

Detailed Test Results:

- Document detailed results of test case execution.
- Include information on passed and failed test cases, along with any deviations.

Defect Report:

- Create a report detailing all logged defects.
- Include information such as defect ID, description, status, severity, and steps to reproduce.

Traceability Matrix:

- Provide a matrix linking test cases to requirements.
- Ensure that all requirements have associated test cases, and vice versa.

Metrics and Trends:

- Include metrics such as test coverage, pass rate, and defect density.
- Identify any trends or patterns observed during testing.

Sign-off:

- Obtain formal sign-off from relevant stakeholders.
- Confirm that the system meets the specified acceptance criteria.

Documentation:

- Update testing documentation, including test cases and test scripts.
- Ensure that all testing artifacts are well-maintained for future reference.

Test Cases:

Functional Test Cases:

- Verify that the homepage loads successfully.
- Test the functionality of the "Sign In" button on the homepage.
- Ensure that the "Sign Up" button works as expected.
- Verify that product categories are correctly displayed on the homepage.
- Test the sorting options on the product listing page.
- Check if the product details page opens when a product is clicked.
- Verify that the "Add to Cart" button adds products to the shopping cart.
- Test the functionality of the shopping cart icon.
- Ensure the "Remove" button in the shopping cart works.
- Verify that product reviews and ratings are displayed correctly.
- Test the "Track Order" functionality.
- Check the "Contact Us" page for proper functionality.
- Verify that links to social media profiles work.
- Test the currency and language selection options.
- Ensure that the website footer links are functional.
- Test the advanced search feature with filters.
- Verify the accuracy of product recommendations on the homepage.
- Test the "Notify Me" feature for out-of-stock products.
- Verify that the "Continue Shopping" button works during checkout.
- Test the accessibility of the website for users with disabilities.

Login Test Cases:

- Verify successful login with valid credentials.
- Test login with an incorrect password.
- Check login with an incorrect username/email.
- Test login with a locked or suspended account.
- Verify the "Forgot Password" link functionality.
- Test login using social media account credentials.
- Check for case sensitivity in usernames and passwords.
- Verify the session timeout behaviour.
- Test login with special characters in the password.
- Test login with a blank username and password fields.
- Ensure the "Remember Me" functionality works as expected.
- Test the "Logout" functionality.
- Check for security measures like CAPTCHA during login.
- Test login on multiple devices and browsers.
- Verify that login attempts are logged for security monitoring.
- Test login with two-factor authentication (if supported).
- Verify login with biometric authentication (e.g., fingerprint, face ID).

- Test login using a one-time password (OTP).

Test Cases for Registration Page

- Verify successful user registration with valid information.
- Test registration with missing required fields (e.g., email, password).
- Check for proper error messages when registration fails.
- Test registration with a pre-existing email or username.
- Verify password strength validation during registration.
- Test registration using special characters in the name and address fields.
- Check if registration data is stored securely.
- Test registration using different browsers.
- Verify that registration confirmation emails are sent.
- Test the redirection after successful registration.

Test Cases for Search Functionality

- Test product search using valid keywords.
- Verify that the search result page displays products.
- Test searching with misspelled keywords and check for suggestions.
- Verify that empty search fields provide a proper message.
- Test searching with special characters in the query.
- Check for case insensitivity in search queries.
- Verify that advanced search filters work as expected.
- Test search with a combination of filters.
- Ensure that the search bar is present on all pages.
- Test searching for out-of-stock products.
- Verify that the search bar works on mobile devices.
- Test the responsiveness of the search results page.
- Check if the search functionality respects user permissions.
- Test search with a very long query string.
- Verify that the "Clear" button in search filters works.

Product Catalogue Test Cases

- Verify that all product categories are listed.
- Test filtering products by category.
- Verify that product images are displayed correctly.
- Test the accuracy of product descriptions.
- Verify that prices are correctly displayed.
- Test sorting products by price.
- Check for product availability indicators.
- Test the "New Arrivals" section for accuracy.
- Verify that product ratings are displayed.
- Test the product comparison feature.
- Check for related product recommendations.
- Test the "Quick View" feature for product details.
- Verify that the catalog page loads within an acceptable time.
- Test the responsiveness of the catalog page.

- Check for accessibility of product information for screen readers.

Shopping Cart Test Cases

- Test adding a product to the shopping cart.
- Verify that the cart icon updates with the item count.
- Test removing a product from the cart.
- Verify that the cart total is correctly calculated.
- Test updating the quantity of items in the cart.
- Check for proper validation of quantity input.
- Verify that discounts and coupons are applied correctly.
- Test the "Continue Shopping" button in the cart.
- Verify that the "Proceed to Checkout" button works.
- Test cart persistence across user sessions.
- Check for a confirmation message after adding/removing items.
- Test cart functionality with both registered and guest users.
- Verify that the cart is emptied after the order is placed.
- Test cart interactions on mobile devices.
- Check for security measures to prevent cart manipulation.

Test Cases for The Checkout Process

- Verify that the checkout process starts from the cart.
- Test selecting a shipping address during checkout.
- Verify that users can add/edit shipping addresses.
- Test selecting a payment method.
- Verify that users can add/edit payment methods.
- Test the application of discounts and coupons.
- Verify the order summary and item details on the checkout page.
- Test the "Place Order" button functionality.
- Verify the display of order confirmation details.
- Test the "Continue Shopping" option after checkout.
- Verify that taxes and shipping costs are accurately calculated.
- Test the ability to review and edit the order before placing it.
- Verify that the checkout process remains secure.
- Test the responsiveness of the checkout page.
- Check for accessibility of checkout steps for users with disabilities.
- Test payment processing using valid credit card details.
- Verify that the payment gateway processes transactions successfully.
- Test payment processing with an invalid credit card number.
- Verify the handling of declined payments.
- Test payment using alternative methods (e.g., PayPal, Apple Pay).
- Verify that payment confirmation emails are sent.
- Test payment processing with international currencies.
- Verify that the billing address matches the payment details.
- Test payment processing while changing the payment method.
- Verify the behaviour of the payment gateway during high traffic.
- Test payment processing on mobile devices.
- Check for security measures like CVV validation.
- Test the cancellation of payment during the checkout process.
- Verify that users receive payment receipts.
- Test payment processing on different browsers.

- Verify the encryption of payment data for security.

Test Cases for Shipping and Delivery

- Verify that the shipping address entered by the user is validated for accuracy and completeness.
- Test entering an incomplete or invalid shipping address to ensure it's rejected.
- Test the selection of different shipping methods during checkout (e.g., standard, express).
- Verify that the estimated delivery dates for each option are accurate.
- Test the accurate calculation of shipping costs based on the chosen shipping method and destination.
- Verify that shipping costs are displayed clearly to the user during checkout.
- Test the functionality to track the status and location of a shipped order.
- Verify that users receive tracking information and updates via email or SMS.
- Test the system's ability to send notifications to users when their package is out for delivery.
- Verify that users receive notifications upon successful delivery.

Boundary and Edge Test Cases

- Password should be between 8-16 characters, should be contain capital letter, numeric value and special character.
- Need to check password with 7,8,15,16,17 character and with and without capital letter, numeric value and special character.

Automation framework:

For framework design I choose Selenium (Java) because:

1.Open-Source

As mentioned earlier, the biggest strength of Selenium is that it is freeware and a portable tool. It has no upfront direct costs involved. The tool can be freely downloaded and its community-based support is freely available.

2.Language Support

Selenium supports various languages, including Java, Perl, Python, C#, Ruby, Groovy, JavaScript, and more. It has its own script, but it is not limited by that language. It can work with various languages – whatever the developers/testers are comfortable with.

3. Supports Operating Systems

Selenium can operate and support multiple Operating Systems (OS) like Windows, Mac, Linux, and UNIX. With the Selenium suite of solutions, a tailored testing suite can be created over any platform and then executed on another one. For instance, you can easily create test cases using Windows OS and run them on a Linux–based system.

4.Support across browsers

Selenium testing offers the advantage of automating web application testing across various browsers: Internet Explorer, Chrome, Firefox, Opera, and Safari, ensuring consistent functionality and reducing manual effort. This becomes highly resourceful while simultaneously executing and testing tests across various browsers.

5.Support for programming languages and framework

Selenium integrates with programming languages and various frameworks. For instance, it can integrate with ANT or Maven type of framework for source code compilation. Further, it can integrate with the TestNG framework for testing applications and reporting purposes. It can integrate with Jenkins or Hudson for Continuous Integration (CI) and even integrate with open-source tools to support other features.

6.Tests across devices

Selenium Test Automation can be implemented for mobile web application automation on Android, iPhone, and Blackberry. This can help in generating necessary results and address issues continuously.

7.Constant updates

Selenium support is community–based, which enables constant updates and upgrades. These upgrades are readily available and do not require specific training. This makes Selenium resourceful and cost-effective as well.

8.Loaded Selenium Suites

Selenium is not just a singular tool or utility, it is a loaded package of various testing tools and so, is referred to as a Suite. Each tool is designed to cater to different testing needs and requirements of test environments.

Selenium can also support Selenium IDE, Selenium Grid, and Selenium Remote Control (RC).

9.Ease of implementation

Selenium offers a user-friendly interface that helps create and execute tests easily and effectively. Its open-source features help users to script their own extensions that make them easy to develop, customize actions, and even manipulate at an advanced level.

Tests run directly across browsers; users can watch while the tests are being executed. Additionally, Selenium's reporting capabilities are one of the reasons for choosing it, as it allows testers to extract the results and take follow-up actions.

10.Re usability and Add-ons

Selenium Test Automation framework uses scripts that can be tested directly across multiple browsers. Concurrently, it is possible to execute multiple tests with Selenium, as it covers almost all aspects of functional testing by implementing add-on tools that broaden the scope of testing.