

TEST PLAN

SauceDemo Web Application

| | |
|-----------------------|-----------------------------------|
| Document Type: | Test Plan |
| Project: | SauceDemo Web Application Testing |
| Version: | 2.0 (Comprehensive) |
| Date: | February 08, 2026 |
| Testing Scope: | Manual & Automated (UI) Testing |
| Status: | Active |

TABLE OF CONTENTS

1. Introduction and Overview
2. SDLC and STLC Overview
3. System Under Test (SUT)
4. Test Scope and Coverage
5. Test Environment and Tools
6. Assumptions and Dependencies
7. Test Scenarios and Test Cases
8. Entry and Exit Criteria
9. Risk Analysis and Mitigation Strategy
10. Traceability Matrix
11. Quality Goals and Key Performance Indicators (KPIs)
12. Test Schedule and Timeline
13. Test Closure and Deliverables
14. Roles and Responsibilities
15. Appendices

1. INTRODUCTION AND OVERVIEW

This comprehensive Test Plan defines the complete testing strategy, scope, resources, risks, and execution approach for both manual and UI automation testing of the SauceDemo web application. The document aligns with industry best practices and course requirements, and is based on actual testing activities performed throughout the project lifecycle.

Purpose

The purpose of this Test Plan is to:

- Define the scope and objectives of testing activities
- Document the testing approach and methodology
- Identify test environments and required resources
- Establish quality criteria and success metrics
- Manage risks and mitigation strategies
- Ensure complete test coverage of critical functionality
- Facilitate effective communication among stakeholders
- Provide a baseline for test progress tracking and reporting

Scope of the Test Plan

This Test Plan encompasses both manual testing and UI automation testing using Selenium WebDriver. It covers the complete user journey from login through checkout, with emphasis on critical business flows. The plan includes 9 automated test cases and multiple manual test scenarios executed across the primary features of the SauceDemo application.

2. SDLC AND STLC OVERVIEW

Software Development Life Cycle (SDLC)

The Software Development Life Cycle (SDLC) is a structured process that guides the development of software from conception through deployment and maintenance. Our project followed a comprehensive SDLC approach with the following phases:

| SDLC Phase | Description |
|-----------------------|--|
| Requirements Analysis | Gathering and documenting functional and non-functional requirements |
| Design | Creating system architecture and test design specifications |
| Development | Coding and implementing application features |
| Testing | Quality assurance through manual and automated testing |
| Deployment | Releasing application to production environment |
| Maintenance | Post-deployment support and defect fixes |

Software Testing Life Cycle (STLC)

The Software Testing Life Cycle (STLC) is a systematic process for planning and executing tests. This project meticulously followed the STLC methodology as outlined below:

| STLC Phase | Implementation in This Project |
|------------------------|--|
| Requirement Analysis | Analyzed SauceDemo functional specifications and user flows |
| Test Planning | Created comprehensive test plan and strategy document |
| Test Case Design | Designed 50+ manual test cases and 9 automated test scenarios |
| Test Environment Setup | Configured Chrome browser, Windows environment, Selenium WebDriver |
| Test Execution | Executed all manual and automated tests in controlled environment |
| Defect Reporting | Documented and reported defects following standardized format |
| Retesting | Performed regression testing after defect fixes |
| Test Closure | Completed testing phase with comprehensive documentation and artifacts |

By following both SDLC and STLC methodologies, we ensured systematic quality assurance throughout the entire project lifecycle, enabling early defect detection and validation of system requirements.

3. SYSTEM UNDER TEST (SUT)

Application Overview

| Attribute | Details |
|------------------|---|
| Application Name | SauceDemo Web Application |
| Application Type | Web-based E-commerce Demo Platform |
| Primary URL | https://www.saucedemo.com |
| Technology Stack | HTML5, CSS3, JavaScript (Frontend) |
| Purpose | Educational demo platform for testing practice |
| User Base | Test automation learners and QA professionals |

Key Features Tested

- User Authentication: Login and logout functionality with valid/invalid credentials
- Product Catalog: Product listing, filtering, and sorting capabilities
- Shopping Cart: Add/remove products, view cart contents, validate cart totals
- Checkout Process: Multi-step checkout flow including information entry and order confirmation
- Product Details: Individual product pages with descriptions, prices, and images
- User Session Management: Maintaining user state throughout shopping experience

Supported Browsers and Platforms

Testing was conducted on:

- Google Chrome (Version 120+)
- Windows 10 Operating System

The application is responsive and accessible across modern browsers, with primary testing focus on Chrome due to its widespread use in the industry.

4. TEST SCOPE AND COVERAGE

In Scope - Manual Testing

The following areas are included in manual testing scope:

- User Login - Valid credentials, invalid credentials, and edge cases
- User Logout - Session termination and state clearing
- Product Catalog - Browsing, filtering, and sorting products
- Product Details - Viewing complete product information
- Add to Cart - Single and multiple product additions
- Remove from Cart - Removing items and updating cart totals
- Cart Verification - Validating cart contents and calculations
- Checkout Process - Complete multi-step checkout flow
- Order Confirmation - Validating order details and confirmation messages
- Error Handling - Application behavior with invalid inputs
- Data Validation - Form validation and error messages

In Scope - UI Automation Testing

UI automation testing covers critical user flows using Selenium WebDriver with Java:

- Test 1: Login with Valid Credentials
- Test 2: Login with Invalid Credentials
- Test 3: Add Product to Cart
- Test 4: Remove Product from Cart
- Test 5: View Cart and Verify Contents
- Test 6: Complete Checkout Process
- Test 7: Verify Order Confirmation
- Test 8: Logout Successfully
- Test 9: Verify Product Sorting Functionality

Out of Scope

The following activities are explicitly excluded from this test plan:

- Performance Testing - Load testing, stress testing, and performance metrics
- Security Testing - SQL injection, XSS vulnerabilities, authentication bypass
- Cross-Browser Testing - Browsers other than Chrome (Firefox, Safari, Edge)
- Mobile Testing - Responsive design testing on mobile devices and tablets
- API Testing - Backend API endpoint testing
- Database Testing - Direct database validation and integrity checks
- Accessibility Testing - WCAG compliance and assistive technology compatibility
- Usability Testing - User experience and interface design evaluation

Test Coverage Summary

| Feature Area | Manual Tests | Automated Tests |
|--------------------|--------------|-----------------|
| Authentication | 6 | 2 |
| Product Management | 8 | 2 |
| Shopping Cart | 12 | 3 |
| Checkout | 15 | 1 |
| Session Management | 4 | 1 |

Total Coverage: 45+ manual test cases, 9 automated test cases covering all critical business flows

5. TEST ENVIRONMENT AND TOOLS

Hardware and Operating System

| Component | Specification |
|------------------|------------------------------------|
| Operating System | Windows 10 (Build 19045 or higher) |
| Processor | Intel Core i5 or higher |
| RAM | Minimum 8 GB |
| Disk Space | Minimum 5 GB free space |

Software and Tools

| Tool/Framework | Version | Purpose |
|--------------------|---------|---------------------------------|
| Google Chrome | 120.0+ | Primary test browser |
| Selenium WebDriver | 4.15+ | UI automation framework |
| Java | JDK 11+ | Automation scripting language |
| JUnit | 4.13+ | Test framework and assertions |
| TestNG | 7.8+ | Test execution and reporting |
| Maven | 3.8+ | Build and dependency management |

Test Data Environment

Test data is sourced from the SauceDemo application's pre-configured demo environment:

- Demo User Accounts: pre-defined usernames and passwords for testing
- Product Inventory: sample products with varying prices and descriptions
- Cart State: ability to reset cart for each test iteration
- No Real Transactions: uses mock checkout process
- Data Isolation: each test run operates independently

6. ASSUMPTIONS AND DEPENDENCIES

Assumptions

The following assumptions have been made for this test plan:

- The SauceDemo environment will remain available and stable throughout testing
- Only non-production test data will be used in all testing activities
- No real financial transactions will be processed during checkout testing
- Internet connectivity will be maintained throughout the testing period
- Chrome browser and required tools will remain installed and functional
- Test environment will not be shared with production or other projects
- Test data will not be modified or corrupted during test execution
- All testers have appropriate access to testing tools and environments

Dependencies

The following dependencies must be satisfied for test execution:

- External Dependencies: SauceDemo application availability and accessibility
- Tool Dependencies: Selenium WebDriver, Java, Chrome WebDriver compatibility
- Data Dependencies: Availability of pre-configured demo user accounts
- Infrastructure Dependencies: Network connectivity and test environment stability
- Resource Dependencies: Testing team availability and tool installation
- Documentation Dependencies: Requirements and design documents for test case creation

Constraints

- Time Constraints: Limited testing window requires prioritization of critical paths
- Environment Constraints: Testing limited to Chrome browser on Windows 10
- Scope Constraints: Performance and security testing excluded from scope
- Budget Constraints: Using free and open-source testing tools

7. TEST SCENARIOS AND TEST CASES

This section details the comprehensive test scenarios and associated test cases designed to validate all critical functionality of the SauceDemo application.

Test Scenario Mapping

| Test Scenario # | Scenario Description |
|-----------------|---|
| TS-001 | User authenticates with valid credentials and accesses dashboard |
| TS-002 | User attempts login with invalid credentials and receives error message |
| TS-003 | User browses product catalog and applies sorting filters |
| TS-004 | User adds single product to shopping cart and verifies cart update |
| TS-005 | User adds multiple products and removes specific items from cart |
| TS-006 | User reviews cart contents and verifies price calculations |
| TS-007 | User completes multi-step checkout process with valid information |
| TS-008 | System displays order confirmation with order details |
| TS-009 | User successfully logs out and session is terminated |
| TS-010 | User sorts products by price and verifies sorting accuracy |

Sample Manual Test Cases

Below are detailed manual test cases for critical scenarios:

Test Case: TC-001 - Login with Valid Credentials

| | |
|--------------------|--|
| Test Case ID | TC-001 |
| Test Scenario | TS-001 |
| Test Title | User Login with Valid Credentials |
| Preconditions | SauceDemo application is accessible and loaded |
| Test Steps | 1. Navigate to https://www.saucedemo.com 2. Enter username "standard_user" 3. Enter password "secret_sauce" 4. Click Login button |
| Expected Result | User is successfully logged in and dashboard is displayed |
| Test Data | Username: standard_user, Password: secret_sauce |
| Priority | Critical |
| Estimated Duration | 5 minutes |

| | |
|--|--|
| | |
|--|--|

Test Case: TC-004 - Add Product to Cart

| | |
|---------------------------|---|
| Test Case ID | TC-004 |
| Test Scenario | TS-004 |
| Test Title | Add Single Product to Shopping Cart |
| Preconditions | User is logged in and viewing product catalog |
| Test Steps | <ol style="list-style-type: none"> 1. Locate product "Sauce Labs Backpack" 2. Click "ADD TO CART" button 3. Verify cart icon shows count 4. Navigate to cart page |
| Expected Result | Product is added to cart, cart count updates, product appears in cart page |
| Test Data | Product: Sauce Labs Backpack (\$29.99) |
| Priority | Critical |
| Estimated Duration | 5 minutes |
| | |

Automated Test Cases

The following 9 automated test cases are executed using Selenium WebDriver and Java:

- TS-001: Test_Login_Valid_Credentials
- TS-002: Test_Login_Invalid_Credentials
- TS-003: Test_Product_Sorting
- TS-004: Test_Add_Product_To_Cart
- TS-005: Test_Remove_Product_From_Cart
- TS-006: Test_Verify_Cart_Contents
- TS-007: Test_Complete_Checkout
- TS-008: Test_Order_Confirmation
- TS-009: Test_Logout_Successfully

8. ENTRY AND EXIT CRITERIA

Entry Criteria

Before commencing testing activities, the following entry criteria must be satisfied:

- Requirement specification document is approved and baseline established
- SauceDemo application is deployed and accessible in test environment
- Test environment (Windows 10, Chrome, Selenium) is properly configured
- All necessary testing tools and software are installed and verified
- Test cases are documented and peer-reviewed for completeness
- Test data (demo accounts, products) is available and prepared
- All testers are trained on tools, procedures, and test cases
- Testing schedule and resources are formally approved
- Risk assessment has been completed and documented

Exit Criteria

Testing can be considered complete when the following exit criteria are met:

- All planned test cases (manual and automated) have been executed
- At least 95% manual test case execution rate achieved
- All 9 automated UI tests execute successfully with passing assertions
- All defects have been identified, logged, and categorized by severity
- All critical and blocker defects have been resolved and verified
- No known defects remain unaddressed in production-critical areas
- Test coverage metrics meet or exceed defined quality goals
- Test documentation and artifacts are complete and reviewed
- Test metrics and KPIs are compiled into final report
- Stakeholder sign-off obtained on test completion status

Suspension/Resumption Criteria

Testing may be suspended if:

- Critical defects prevent continuation of test execution
- Test environment becomes unstable or inaccessible
- Required testing resources become unavailable
- Critical requirements remain unmet or unclear

Testing will resume once blocking issues are resolved.

9. RISK ANALYSIS AND MITIGATION STRATEGY

This section identifies potential risks to the testing process and outlines mitigation strategies to minimize their impact on project success.

Risk Assessment Matrix

Risks have been categorized by probability and impact, resulting in overall risk level:

| Risk ID | Risk Description | Probability | Impact | Level |
|---------|--|-------------|----------|--------|
| R-001 | Critical login functionality fails | Low | Critical | HIGH |
| R-002 | Test environment becomes unavailable | Medium | High | MEDIUM |
| R-003 | Demo data inconsistency or corruption | Medium | Medium | MEDIUM |
| R-004 | Inadequate test data or sample products | Low | Medium | LOW |
| R-005 | Chrome browser/driver incompatibility issues | Low | High | MEDIUM |
| R-006 | Tester resource unavailability | Very Low | High | LOW |

High-Risk Area: Critical User Flows

Risk: Core user flows (login, checkout) failure

Mitigation Strategy:

- Prioritize manual testing of authentication and checkout processes
- Develop comprehensive automated test coverage for critical paths
- Conduct daily smoke tests to detect regressions early
- Maintain detailed test logs for post-mortem analysis
- Establish communication protocol for immediate issue escalation

Medium-Risk Area: Environment Stability

Risk: Demo data inconsistency or environment unavailability

Mitigation Strategy:

- Implement flexible test assertions to handle minor data variations
- Document baseline environment configuration for recovery
- Use try-catch blocks and error handling in automation scripts
- Maintain backup test data and recovery procedures

- Monitor environment status continuously during test execution

Low-Risk Area: UI Changes

Risk: Minor UI changes in demo environment

Mitigation Strategy:

- Focus locators on stable, semantic HTML elements
- Use flexible locator strategies (CSS selectors with multiple attributes)
- Regular review and update of automation scripts
- Maintain documentation of known UI quirks and workarounds

10. TRACEABILITY MATRIX

Requirements traceability ensures every requirement is tested and every test is linked to a requirement. This prevents scope creep and ensures complete coverage of functional specifications.

| Requirement ID | Requirement Description | Test Scenario | Automated Test |
|----------------|--|---------------|----------------|
| REQ-001 | User authentication with valid credentials | TS-001 | Yes - TS-001 |
| REQ-002 | User authentication with invalid credentials | TS-002 | Yes - TS-002 |
| REQ-003 | Product listing and filtering | TS-003 | Yes - TS-003 |
| REQ-004 | Add product to cart | TS-004 | Yes - TS-004 |
| REQ-005 | Remove product from cart | TS-005 | Yes - TS-005 |
| REQ-006 | Cart validation and totals | TS-006 | Yes - TS-006 |
| REQ-007 | Complete checkout process | TS-007 | Yes - TS-007 |
| REQ-008 | Order confirmation display | TS-008 | Yes - TS-008 |
| REQ-009 | User logout functionality | TS-009 | Yes - TS-009 |
| REQ-010 | Product sorting capability | TS-010 | Yes - TS-010 |

Traceability Coverage: 100% of documented requirements have corresponding test scenarios and automated tests.

11. QUALITY GOALS AND KEY PERFORMANCE INDICATORS (KPIs)

Quality Objectives

The primary quality objectives for this testing initiative are:

- Achieve comprehensive functional test coverage of all critical user paths
- Identify and document all defects prior to production release
- Validate system performance meets minimum acceptable standards
- Ensure user experience meets stakeholder expectations
- Establish repeatable automated testing processes for regression testing

Key Performance Indicators (KPIs)

| KPI | Target | Success Criteria |
|----------------------------|--------|---|
| Test Case Execution Rate | ≥ 95% | At least 95% of planned tests executed |
| Manual Test Coverage | ≥ 90% | Coverage of all specified manual test scenarios |
| Automated Test Pass Rate | 100% | All 9 automated tests pass successfully |
| Defect Detection Rate | ≥ 80% | Identify at least 80% of existing defects |
| Critical Defect Resolution | 100% | All critical/blocker defects resolved |
| Test Schedule Adherence | ≥ 95% | Complete testing within planned timeline |
| Automation Code Coverage | ≥ 85% | Automated tests cover 85% of critical paths |
| Defect Escape Rate | ≤ 5% | No more than 5% of defects escape to production |

Quality Metrics

The following metrics will be tracked throughout the testing lifecycle:

- Test Execution Progress: % of test cases executed vs. planned
- Defect Metrics: Total defects, severity distribution, status trends
- Test Effectiveness: Defect detection rate, defect density per module
- Test Efficiency: Test cases executed per day, effort vs. planned
- Schedule Performance: Actual vs. planned testing timeline
- Automation Metrics: Automated vs. manual test ratio, maintenance effort

12. TEST SCHEDULE AND TIMELINE

This section outlines the planned testing timeline and resource allocation.

Testing Timeline

| Phase | Duration | Start Date | End Date |
|-------------------------------|----------|------------|----------|
| Test Planning & Preparation | 2-3 days | Day 1 | Day 3 |
| Manual Test Case Design | 3-4 days | Day 2 | Day 5 |
| Automation Script Development | 4-5 days | Day 3 | Day 8 |
| Manual Test Execution | 5-6 days | Day 6 | Day 12 |
| Automation Test Execution | 3-4 days | Day 9 | Day 13 |
| Defect Analysis & Retesting | 4-5 days | Day 11 | Day 16 |
| Test Closure & Reporting | 2-3 days | Day 14 | Day 17 |

Daily Testing Schedule

Daily testing activities follow this structure:

- 8:00 AM - 8:30 AM: Daily standup and progress review
- 8:30 AM - 12:00 PM: Test execution (manual and automated)
- 12:00 PM - 1:00 PM: Lunch break
- 1:00 PM - 4:30 PM: Continue test execution and defect logging
- 4:30 PM - 5:00 PM: Defect analysis and EOD reporting
- 5:00 PM - 5:30 PM: Documentation and preparation for next day

13. TEST CLOSURE AND DELIVERABLES

Test Closure Activities

Upon completion of all testing activities, the following closure activities are performed:

- Execute final smoke tests to confirm system stability
- Review and finalize all test execution reports
- Compile comprehensive defect summary and metrics
- Document lessons learned and improvement recommendations
- Archive test cases, scripts, and supporting documentation
- Conduct post-project review with stakeholders
- Obtain formal sign-off on testing completion

Deliverables

The following deliverables will be provided upon test completion:

- Test Plan Document: Comprehensive testing strategy and approach (this document)
- Test Cases: Detailed manual test cases with steps, data, and expected results
- Automation Scripts: Java-based Selenium WebDriver test scripts with Maven configuration
- Test Execution Report: Summary of test results, pass/fail status, and execution metrics
- Defect Report: Detailed defect log with descriptions, severity, and resolution status
- Test Metrics & Analytics: KPI metrics, coverage analysis, and trend reports
- Lessons Learned Document: Recommendations for process improvement and best practices
- Test Artifacts: Screenshots, logs, and supporting documentation
- Requirements Traceability Matrix: Mapping of requirements to test cases

Success Criteria for Test Closure

- All entry criteria for test closure are satisfied
- Test execution rate meets or exceeds 95%
- All critical and blocker defects are resolved
- Test metrics meet defined quality goals
- All documentation is complete and reviewed
- Stakeholder approval and sign-off obtained

14. ROLES AND RESPONSIBILITIES

| Role | Responsibility | Key Activities |
|---------------------|---|--|
| Test Manager | Overall testing strategy and coordination | Planning, scheduling, reporting, stakeholder communication |
| QA Lead | Test case design and quality standards | Test design, execution oversight, defect prioritization |
| Manual QA Tester | Execute manual test cases | Test execution, defect logging, issue documentation |
| Automation Engineer | Develop and maintain automation scripts | Script development, execution, maintenance and debugging |
| Product Owner | Requirements validation and approval | Requirement clarification, acceptance criteria definition |

Communication Plan

Testing progress and issues will be communicated as follows:

- Daily Standup: 15-minute synchronous meeting with full team
- Weekly Status Report: Comprehensive metrics and progress summary
- Defect Alerts: Immediate notification for critical/blocker issues
- Milestone Reviews: Gateway reviews at major phase completions
- Final Report: Comprehensive testing summary and recommendations

15. APPENDICES

Appendix A: Demo User Accounts

| Username | Password | Account Type |
|-------------------------|--------------|------------------------------|
| standard_user | secret_sauce | Standard User |
| locked_out_user | secret_sauce | Locked Account |
| problem_user | secret_sauce | User with Sorting Issues |
| performance_glitch_user | secret_sauce | User with Performance Issues |

Appendix B: Glossary of Terms

- API: Application Programming Interface
- CSS: Cascading Style Sheets
- HTML: HyperText Markup Language
- KPI: Key Performance Indicator
- QA: Quality Assurance
- SDLC: Software Development Life Cycle
- STLC: Software Testing Life Cycle
- SUT: System Under Test
- UI: User Interface
- XPath: XML Path Language for locating elements

Appendix C: References and Resources

- SauceDemo Application: <https://www.saucedemo.com>
- Selenium WebDriver Documentation: <https://www.selenium.dev/documentation/>
- Java TestNG Framework: <https://testng.org/doc/>
- Maven Project Management: <https://maven.apache.org/>
- Chrome WebDriver: <https://chromedriver.chromium.org/>

Appendix D: Test Execution Checklist

- Environment Setup: Chrome browser, WebDriver, Java, Maven installed and configured
- Data Preparation: Demo accounts and test data verified and ready
- Tool Configuration: Selenium WebDriver paths and configurations validated
- Test Case Review: All test cases reviewed and understood by testers
- Baseline Documentation: Initial system behavior documented
- Resource Allocation: All team members assigned and available
- Communication: All stakeholders informed of test start date
- Defect Tracking: Defect tracking system configured and accessible
- Documentation: Templates for logs and reports prepared

CONCLUSION

This comprehensive Test Plan provides a structured approach to validating the SauceDemo web application through both manual testing and UI automation. By following this plan, we ensure systematic quality assurance, complete functional coverage, and early defect detection.

The combination of manual test cases and automated scripts provides comprehensive coverage while maintaining efficiency. Risk-based prioritization ensures critical functionality is thoroughly tested, and the defined KPIs establish clear success criteria for the testing initiative.

Regular progress tracking, defect management, and adherence to the defined SDLC/STLC methodology will ensure successful test execution and delivery of a high-quality application ready for production use.

For questions or clarifications regarding this Test Plan, please contact the Test Manager or QA Lead.