**WIPRO JAVA SELENIUM – BATCH-10, TEAM-06** 

Change Log

| **Version** | **Change Date** | **By** | **Description** |
| --- | --- | --- | --- |
| version number | Date of Change | Name of person who made changes | Description of the changes made |
|  |  |  |  |
|  |  |  |  |

**TABLE OF CONTENTS**

**1** **INTRODUCTION** 3

1.1 Scope 3

1.1.1 In Scope3

1.1.2 Out of Scope3

1.2 Quality Objective 4

1.3 Roles and Responsibilities 4

**2** **TEST METHODOLOGY** 5

2.1 Overview 5

2.2 Test Levels 5

2.3 Bug Triage 5

2.4 Suspension Criteria and Resumption Requirements 6

2.5 Test Completeness 6

**3** **TEST DELIVERABLES** 7

**4** **RESOURCE & ENVIRONMENT NEEDS** 8

4.1 Testing Tools 8

4.2 Test Environment 9

**5** **TERMS/ACRONYMS** 11

# 1 Introduction

The purpose of this test plan is to define the testing strategy, objectives, scope, and responsibilities for the **BStackDemo e-commerce application**. This plan includes both manual and automation testing to ensure the application’s functionality, usability, and reliability.

BStackDemo is a sample online store used for testing practice. It includes modules such as login, product search, product filters, sorting, cart management, checkout flow, PDF download of orders, and navigation to features like Orders, Offers, and Favourites.

**1.1 Scope**

### **1.1.1 In Scope**

The following areas are covered as part of the testing activities for BStackDemo:

* **Login & Logout**: Validation with demo user accounts.
* **Product Search**: Searching by full/partial product names.
* **Filters & Sorting**: Applying vendor filters (Apple, Samsung) and sorting products by price (lowest to highest, highest to lowest).
* **Product Browsing**: Viewing product list, product details.
* **Cart Management**: Adding/removing items, updating quantity, verifying subtotal.
* **Checkout Flow**: Shipping details, order confirmation, PDF invoice download.
* **Navigation Links**: Orders, Favourites, Offers pages.
* **Cross-browser Testing**: Validation on Chrome, Firefox, Edge.
* **UI Validation**: Buttons, dropdowns, cart popups, confirmation messages.
* **Data-driven Testing**: Using Excel/TestNG DataProviders for search/filter inputs.

#### 1.1.2 Out of Scope

* Real payment gateway transactions (checkout flow is demo only).
* Backend admin modules, analytics, or reporting features.
* Mobile application testing (focus is on desktop browsers).
* Performance/load testing (covered in separate performance test plan if needed).

**1.2 Quality Objectives**

* Ensure core e-commerce workflows (login, cart, checkout) function as expected.
* Validate search and filter/sort produce correct results.
* Confirm UI consistency across supported browsers.
* Enable automation coverage for regression testing.
* Identify and report defects early to minimize risks.
* Maintain high test coverage aligned with requirements.

**1.3 Roles and Responsibilities**

| **Role** | **Responsibilities** |
| --- | --- |
| Test Manager | Approves the test plan, allocates resources, monitors overall progress. |
| Manual Tester | Executes manual test cases, validates UI, logs defects, verifies fixes. |
| Automation Tester | Creates Selenium + TestNG automation scripts, maintains regression suite. |
| Business Analyst | Provides functional requirements, clarifies scenarios, ensures coverage. |
| QA Lead | Oversees execution progress, ensures adherence to standards, coordinates with developers. |
| Developer | Resolves defects, supports testers during debugging, confirms fixes in staging. |
| Reporting Lead | Prepares execution reports, maintains test documentation, updates defect logs. |
| Product Owner | - Prioritize features and bug fixes  - Approve test sign-offs |

# 2 Test Methodology

## 2.1 Overview

Testing for BStackDemo will follow the Agile Testing Methodology, aligning with iterative sprints and continuous integration. Both manual testing and automation testing will be used.

* **Automation**: Selenium WebDriver, TestNG, Java, Maven.
* **Manual**: Functional validation of search, filter, cart, checkout, and navigation.
* **Data-driven testing**: Using Excel and TestNG DataProviders for varied inputs.

## 2.2 Test Levels

Test levels define the stages and types of testing to ensure comprehensive quality checks at various points of the software delivery cycle. The selection and scope of test levels depend on factors such as project scope, time, resources, and budget.

Typical test levels for this project:

* **Unit Testing**: Conducted by developers on small modules (e.g., product add-to-cart logic).
* **Integration Testing**: Validating interactions between modules (e.g., search → product list → cart).
* **System Testing**: End-to-end workflows using Selenium automation.
* **User Acceptance Testing (UAT)**: Stakeholder review of test cases and results to validate business flow correctness.

Depending on constraints, exploratory and regression testing are integrated at relevant stages to ensure ongoing product stability and discovery of edge-case defects.

## 2.3 Bug Triage

The bug triage process ensures efficient handling of defects discovered during testing by:

* Classify bugs by **severity** (Critical, High, Medium, Low) and **priority** (P1–P4).
* Critical bugs (e.g., cart not updating, checkout blocked) must be fixed immediately.
* Weekly triage meetings to review open defects and update status.
* Bugs tracked via **Jira/GitHub Issues**

Goal: To maintain a focused and prioritized defect backlog that ensures critical bugs are fixed timely, preventing delays or risks to release schedules.

## 2.4 Suspension Criteria and Resumption Requirements

* Suspension Criteria: Define situations where testing must be halted temporarily, such as:
  + Test environment or website unavailable.
  + Blocking defects in login or checkout flow.
  + Major build instability preventing test execution.
* Resumption Requirements: Conditions that must be met for testing to continue include:
  + Test environment restored.
  + Blocking issues fixed and retested.
  + Stable build deployed with valid test data.

This ensures testing resources focus on productive testing and helps avoid time waste due to external, unresolved issues.

## 2.5 Test Completeness Criteria

Testing is considered complete when predefined criteria are satisfied, such as:

* 100% Test Coverage: All identified test cases aligned with requirements and user stories have been executed.
* Execution of All Manual and Automated Tests: Both manual exploratory and automated regression suites have run with acceptable results.
* Bug Resolution: All open critical and high-severity defects are fixed, retested, and closed. Remaining low-priority defects are documented with decisions to fix later.
* Sign-offs: Formal approvals from product owners and QA stakeholders confirming that the system meets agreed quality standards and is ready for release.
* Non-Functional Checks: Performance, security, and usability tests have met the acceptance criteria without major issues

# 3 Test Deliverables

This document outlines the detailed test deliverables that will be provided throughout the testing lifecycle for the [**https://bstackdemo.com/**](https://bstackdemo.com/) sample e-commerce application. These deliverables ensure structured planning, quality execution, defect tracking, and final approval of the tested product.

| Deliverable | Description | Relevance to <https://bstackdemo.com/> |
| --- | --- | --- |
| Test Plan | Defines the scope, resources, schedule, and approach of testing. | Ensures clear objectives for modules like product listing, filtering, login, add-to-cart, and checkout. |
| Test Cases | Step-by-step scenarios to validate application behavior. | Covers user flows like product search, sorting, filtering, adding items to cart, login, and checkout. |
| Requirement Traceability Matrix (RTM) | Maps requirements to test cases to ensure coverage. | Ensures all functional requirements (UI, APIs, DB validations) are fully validated. |
| Bug Reports | Logs of issues identified during testing with severity and status. | Tracks defects in login flow, product selection, cart management, checkout, and payment handling. |
| Test Strategy | High-level approach including tools, environment, and types of testing. | Specifies Selenium/WebDriver for UI automation and Postman/REST Assured for API validation. |
| Test Metrics | Quantitative data such as test coverage, defect density, and pass/fail ratio..   |  | | --- | | Measures readiness for release and highlights areas needing improvement in workflows. |
| Customer Sign Off | Formal approval from the client post testing. | Confirms that all workflows of bstackdemo.com (browse, login, purchase, checkout) meet expected quality standards. |

**4 Resource & Environment Needs**

**4.1 Testing Tools**

**4.1.1 Requirements Tracking Tools:**

| **Tool** | **Type** | **Purpose** | **User Purpose** |
| --- | --- | --- | --- |
| Jira | Agile Requirements & Issue Tracker | Track epics, user stories, requirements, tasks | Business Analyst, PO |
| Confluence | Documentation & Collaboration Space | Document and manage requirement specs | Business Analyst, Dev |

**4.1.2.Bug Tracking Tools:**

| **Tool** | **Type** | **Purpose** | **User Purpose** |
| --- | --- | --- | --- |
| Jira Issues | Integrated Bug & Requirement Tracker | Report and track defects, link to stories | QA Engineers, Developers |
| MantisBT | Standalone Bug Tracking System | Track and manage defects independently | QA Team |

**4.1.3 Automation Tools:**

| **Tool** | **Type** | **Purpose** | **User Purpose** |
| --- | --- | --- | --- |
| Selenium + TestNG | Open Source | Cross-browser UI automation for regression testing | Frontend Dev + QA |

**4.2 Test Environment**

## Test Environment Requirements

### **A. Hardware**

* **Servers:**
  + **Minimal**: 4 GB RAM, 1 CPU core, SSD storage (≥ 30 GB)
  + **Recommended**: 8 GB RAM, 4 CPU cores (Intel Core i5/E5-class or equivalent), fast SSD storage (≥ 100 GB)
* **Test Workstations:**
  + Modern desktop or laptop with at least 8–16 GB RAM
  + Dual-core CPU (i5 or equivalent), SSD
  + Screens with resolution 1280×1024 or higher

### **B. Software**

* **Operating Systems:**
  + Windows 8 or newer (Windows 10/11)—for desktops and test workstations
  + Server OS (e.g. Windows Server 2012+, Linux variants) for staging environments
* **Office & Productivity Tools:**
  + Microsoft Office 2013 or above for documentation
* **Email Client:**
  + Microsoft Exchange client—for validating notification workflows
* **Browsers :**
  + Latest stable versions of Chrome, Firefox, and Edge
* **Supporting Software:**
  + Relevant runtime environments, such as .NET Framework, Java JDKs, or middleware, to mirror production configuration

**C. Network & Infrastructure**

* Dedicated local or virtual network setup that closely resembles production network topology, including firewalls, DNS, VPN, and load balancers
* Stable internet access; bandwidth and latency similar to expected production usage
* **D. Setup Process & Validation Workflow**

1. **Design Environment**: Define hardware/software specs, required tools and data.
2. **Provision and Configure**: Build staging servers, install operating systems and browsers, set up Exchange clients, etc.
3. **Access Control**: Assign testers and administrators to the environment.
4. **Smoke Testing**: Execute initial sanity tests to ensure environment stability and connectivity
5. **Maintain Environment**: Regular updates, version control, issue logging, and availability monitoring

# 5 Terms / Acronyms

Here's a detailed **Terms/Acronyms** section tailored for your **Test Strategy Document** for [Acko Insurance](http://www.acko.com/) under the **Agile methodology**:

| **TERM / ACRONYM** | **DEFINITION** |
| --- | --- |
| **API** | Application Programming Interface – A set of protocols and tools for building and interacting with software applications. Used extensively for Acko’s backend integrations. |
| **AUT** | Application Under Test – Refers to the Acko Insurance website or mobile app currently being tested. |
| **UI** | User Interface – The front-end interface where users interact with the Acko platform. |
| **UX** | User Experience – Overall user satisfaction and usability of Acko’s digital offerings. |
| **POM** | Page Object Model – A test automation design pattern used in Selenium for UI testing. |
| **TC** | Test Case – A set of steps executed to verify a specific functionality. |
| **TDD** | Test Driven Development – A software development process where tests are written before the code. |
| **BDD** | Behavior Driven Development – Development approach that encourages collaboration between QA, developers, and business stakeholders. |
| **UAT** | User Acceptance Testing – Final testing phase to ensure the application meets business requirements. |
| **SLA** | Service Level Agreement – A contract defining the level of service expected from a vendor or API. |
| **JIRA** | A project management and issue tracking tool used for Agile teams. |
| **CI/CD** | Continuous Integration / Continuous Deployment – Practice of automating code integration and delivery to ensure faster, safer releases. |
| **RTM** | Requirements Traceability Matrix – A document mapping requirements to their corresponding test cases. |
| **SRS** | Software Requirements Specification – A document that captures functional and non-functional requirements. |
| **BRD** | Business Requirements Document – A formal document that describes the business needs and requirements. |
| **STLC** | Software Testing Life Cycle – A sequence of specific activities conducted during testing. |
| **SDLC** | Software Development Life Cycle – A framework that defines tasks performed at each stage in the software development process. |
| **AGILE** | A flexible, iterative software development methodology promoting continuous collaboration and testing. |
| **DEFECT / BUG** | An issue in the system that causes it to behave unexpectedly or incorrectly. |