Software Test Plan (STP)

Project: E-Commerce Software(SnapBuy)
Version: 1.0
Authors: <Neha PM (PES1UG23AM183)> <Raksha (PES1UG23AM229)> <Rakshitha M (PES1UG23AM231)>
Date: 14-09-2025
Status: Draft

1. Introduction

Purpose: This document defines the test plan for the SnapBuy E-Commerce Platform v1.0. It outlines the objectives, scope, strategy, resources, schedule, and responsibilities for all testing activities to ensure the system meets its specified requirements.

Scope: Testing covers all in-scope features from the SRS, including user authentication, product catalog, shopping cart, checkout and payment processing, order management, notifications, and admin functionalities. Third-party logistics, warehouse automation, and vendor onboarding are explicitly out of scope.

References: SnapBuy SRS v1.0, UI/UX Design Mockups, GDPR Guidelines, PCI-DSS

Definitions: SRS(Software Requirements Specification), RTM(Requirements Traceability Matrix), UAT(User Acceptance Testing), PCI-DSS(Payment Card Industry Data Security Standard), SKU(Stock Keeping Unit), COD(Cash on Delivery)

2. Test Items

- User Authentication Module
- Product Catalog & Search Module
- Shopping Cart Module
- Checkout & Payment Module
- Order Management Module
- Notification Service Module
- Admin Dashboard Module

3. Features to be Tested

Features mapped to SRS requirement IDs:

- FR-001: User Registration & Login
- FR-002, FR-003: Product Browsing & Search with Filters
- FR-004, FR-005: Shopping Cart Management & Calculation

- FR-006, FR-007, FR-008: Secure Checkout, Multiple Payment Modes, Order Confirmation
- FR-009, FR-010: Order Tracking & Viewing Order History
- FR-011: Email/SMS Notifications
- FR-012, FR-013: Admin Product & Inventory Management
- FR-015: Admin Sales Reports
- NFR-001: Search Performance (<2s response)
- NFR-002: System Reliability (99% uptime)
- SR-001, SR-002, SR-003, SR-004, SR-005: All Security Requirements

4. Features Not to be Tested

- Third-party logistics system integrations (e.g., actual shipping carrier APIs).
- Underlying code of third-party payment gateways (Razorpay/PayPal sandbox functionality will be tested, but not their internal code).
- External email/SMS service provider internals (e.g., Twilio, Firebase Cloud Messaging). Their APIs will be mocked or used in sandbox mode for validation.
- Server hardware and operating system stability (assumed to be managed by hosting provider).

5. Test Approach / Strategy

Levels:

- Unit Testing: (Developer responsibility) Testing individual components/functions.
- Integration Testing: Testing interactions between modules
- System Testing: End-to-end testing of the complete application from the user's perspective.
- Acceptance Testing (UAT): Testing by end-users to validate readiness for production.

Types:

- Functional Testing: Validating all functional requirements.
- Regression Testing: Ensuring new changes don't break existing functionality.
- Performance Testing: Load and stress testing on critical paths like search and checkout.
- Usability Testing: Assessing the user interface for clarity and ease of use.
- Security Testing: Validating authentication, authorization, data encryption, and PCI compliance.

Entry Criteria: Requirement baselined and SRS v1.0 approved, Testable build is available in the designated test environment, Test data is prepared and available, Test cases are written and reviewed

Exit Criteria: 100% of planned test cases executed, All critical and major defects are closed, Requirement coverage is 100% as per RTM, UAT is signed off by the Product Owner

5.1 Security Validation

- Validate password is hashed (not plain text) in the database.
- Verify all pages, especially login and checkout, use HTTPS (TLS 1.2+).
- Test for SQL Injection and XSS vulnerabilities in search and login fields.
- Validate OTP flow for high-value transactions (if implemented per SR-003).
- Confirm session timeout after 10 minutes of inactivity.
- Test role-based access control (e.g., customer cannot access admin dashboard).

6. Test Environment

Hardware: Cloud-based VM (e.g., AWS EC2) or local server with specified RAM/CPU, Windows/Mac PCs, iOS/Android mobile devices.

Software:

- OS: Windows 11 / macOS Sonoma / Android 13 / iOS 16
- Browsers: Chrome, Firefox, Edge, Safari
- Backend: Node.js/Python/Java Application Server, MySQL/PostgreSQL Database
- Payment Sandbox: Razorpay/PayPal sandbox integration
- Notification Sandbox: Fake SMTP server (e.g., MailHog), SMS sandbox

Tools: Jira, Trello, or Excel/Sheets(RTM & test cases), Selenium WebDriver (UI), Postman (API), Cypress (UI)**,** JMeter or Lighthouse**,** GitHub Issues

Test Data: Dummy user accounts (customers, admin)**,** Dummy product catalog with various categories, prices, and stock levels, Test payment cards (success, failure, invalid scenarios), Test email addresses and phone numbers for notifications.

7. Test Schedule

Milestones:
- Test case design: 15-Sep-2025
- Environment setup: 17-Sep-2025
- Test execution start: 18-Sep-2025
- Test execution end: 30-Sep-2025
- UAT: 02-Oct-2025 to 05-Oct-2025

8. Test Deliverables

- Test Plan (this document)
- Test Cases (manual & automated)
- Test Scripts
- Test Data
- Test Execution Logs
- Defect Reports
- Test Summary Report

9. Roles and Responsibilities

Role	Name	Responsibility
QA Lead	<Neha PM>	Prepare plan, coordinate execution
Test Engineer	<Raksha>	Design & execute test cases, log defects
Developer	<Rakshitha M>	Support defect fixes and triage
Product Owner	<Prof. Arpitha K>	Approve test results, sign-off readiness

10. Risks and Mitigation

Risk	Mitigation
Delay in stable build delivery	Schedule regular build deployments. Use a CI/CD pipeline for automation.
Test data setup is complex	Automate test data creation scripts. Use database snapshots
Unavailability of third-party sandboxes	Develop and use mock services for critical dependencies.

11. Assumptions & Dependencies

- The development team will provide a stable build for testing as per the schedule.
- Access to payment gateway and notification service sandboxes will be provided.
- The test environment will be isolated and stable for the duration of testing.12. Suspension & Resumption Criteria

Suspend testing if:

- The test environment is down for more than 4 working hours.
- A critical defect is found that blocks testing of a major feature (e.g., users cannot login, checkout is broken).

Resume testing after:

- The test environment is restored and verified.
- The blocking critical defect is fixed and a new build is deployed.

13. Test Case Management & Traceability

RTM ensures mapping of SRS requirements to test cases.
Example:
FR-001 (Login) → TC-LOGIN-01 (Valid Login), TC-LOGIN-02 (Invalid Login)

FR-004 (Add to Cart) → TC-CART-01 (Add Item), TC-CART-02 (Remove Item)

NFR-001 (Performance) → TC-PERF-01 (Search Response Time <2s)

SR-001 (Password Hashing) → TC-SEC-01 (DB Password Audit)

14. Test Metrics & Reporting

Metrics collected:
- % test cases executed
- % passed/failed
- Defect density
- Defect aging
- Requirement coverage

Reports:
- Daily execution status
- Final Test Summary Report

15. Approvals

Role	Name	Signature / Date
QA Lead	Neha PM	neha - 14.09.2025
Dev Lead	Rakshitha M	rakshitha - 14.09.2025
Product Owner	Prof. Arpitha K	