Software Test Plan (STP) – Online Food Delivery System

Project: Food Delivery Service System
Version: 1.0
Authors: <QA Team>
Date: 14-09-2025
Status: Draft

1. Introduction

Purpose: This document defines the test plan for the Food Delivery Service System v1.0. It outlines objectives, scope, strategy, resources, schedule, and responsibilities for testing. The purpose is to validate system functionalities for customers, restaurants, delivery agents, and administrators while ensuring performance, security, usability, and scalability.

Scope: Testing covers core features like customer registration, menu browsing, order placement, payments, delivery tracking, restaurant management, and analytics reporting. Integration with external APIs (payment gateways, SMS/email services, GPS) will be verified. Third-party food preparation and infrastructure maintenance are excluded.

References: OFDS SRS v1.0, Design Specifications v1.0, Banking Standards, PCI-DSS.

Definitions: API: Application Programming Interface PCI-DSS: Payment Card Industry Data Security Standard   GPS: Global Positioning System

2. Test Items

- Customer registration & OTP verification
- Restaurant and dish browsing with filters
- Cart management (add/remove items, quantity selection)
- Order placement and tracking
- Payment gateway integration
- Delivery agent navigation and status updates
- Restaurant dashboard for menu updates and analytics
- Admin reporting and monitoring tools
- System performance, uptime, and security features

3. Features to be Tested

The features to be tested are mapped to their corresponding SRS requirement IDs:

- FDS-F-001: Customer account creation and OTP verification
- FDS-F-002 / 003: Search and filter for restaurants and dishes
- FDS-F-004 / 009: End-to-end order placement workflow, including cart management, address selection, coupon application, and secure payment
- FDS-F-010: Real-time order and delivery tracking with ETA display
- FDS-F-011 / 013: Restaurant ability to accept/reject orders and manage menu items and pricing
- FDS-F-012: Delivery agent order assignment and route navigation
- FDS-F-015: Admin order report generation
- FDS-NF-002: Payment processing speed (80% of payments within 3 seconds)
- FDS-NF-003: System scalability and performance under high load (up to 50,000 concurrent users)
- FDS-SLA-001: System availability (99.9% uptime)

4. Features Not to be Tested

- Internal backend logic and processing of third-party vendors (e.g., Stripe, PayPal, Google Maps).
- The actual food preparation process by restaurants.
- Firmware and hardware of delivery agent smartphones or restaurant devices.

5. Test Approach / Strategy

Levels:

1. Unit Testing: Verify individual functionalities like OTP verification, cart updates, and address inputs.
2. Integration Testing: Test interactions between modules and APIs (payment, notifications, GPS).
3. System Testing: End-to-end workflows like registration → order placement → tracking.
4. Acceptance Testing (UAT): Verify business requirements from the customer and admin perspectives.

Types:

1. Functional testing
2. Regression testing
3. Performance testing (load, stress, latency)
4. Security testing (MFA, encryption, data access)
5. Usability testing
6. API validation testing
7. Compatibility testing across devices and browsers

Entry Criteria:

1. Stable build v1.0 deployed
2. APIs connected with sandbox/test accounts
3. Test data provisioned for users, restaurants, and delivery agents
4. Environment setup validated

Exit Criteria:

1. All planned test cases executed
2. No critical defects open
3. Business requirements verified by stakeholders

5.1 Security Validation

1. Validate that user credentials, OTPs, and payment information are transmitted securely via HTTPS
2. Verify compliance with PCI-DSS for all payment transactions
3. Validate GDPR compliance for customer data privacy
4. Test Multi-Factor Authentication for admin and delivery partners
5. Perform fuzz testing on inputs like OTP, payment details, and addresses
6. Simulate SQL injection, brute-force attempts, and DDoS conditions

6. Test Environment

Hardware:

1. Mobile phones with GPS for delivery agents
2. Web browsers on laptops/desktops
3. Cloud servers with API endpoints

Software:

1. Food Delivery Service v1.0
2. Payment API sandbox accounts
3. SMS/email APIs
4. Google Maps API for tracking

Tools:

1. Postman (API testing)
2. JMeter (load testing)
3. Selenium (UI automation)
4. OWASP ZAP (security scanning)
5. Jira (defect management)

Test Data:

1. Pre-configured customer accounts
2. Dummy restaurants and menu items
3. Payment methods with test credentials
4. GPS coordinates for tracking simulation

7. Test Schedule

Milestone	
Test case design completed	18-09-2025
Test environment setup	20-09-2025
Test execution starts	21-09-2025
Test execution ends	05-10-2025
UAT	07-10-2025 to 10-10-2025

8. Test Deliverables

- Software Test Plan (this document)
- Test Cases (manual and automated)
- Test Scripts
- Test Data sets
- Execution Logs
- Security Test Reports
- Performance Test Reports
- Defect Reports
- Final Test Summary Report

9. Roles and Responsibilities

Role	Name	Responsibility
QA Lead	Darshan Gowda BM	Prepare plan, coordinate execution
Test Engineer	Chirag BL	Design & execute test cases, log defects
Developer	Jahnavi	Support defect fixes and triage

Product Owner	Isha	Approve test results, sign-off readiness

10. Risks and Mitigation

Risk	Mitigation
Delay in payment API sandbox access	Engage providers early and prepare mocks
GPS tracking failures in offline scenarios	Use controlled simulation environments
Heavy traffic during peak testing	Engage vendor early and maintain test stubs

11. Assumptions & Dependencies

1. Payment APIs will be available with test credentials
2. GPS services will simulate tracking data during tests
3. Test data will include at least 100 customers, 50 restaurants, and 30 delivery agents
4. Admin console access will be granted with required permissions
5. Compliance standards will be provided for reference by the security team

12. Suspension & Resumption Criteria

Suspend testing if:

- API endpoints are unavailable for more than 2 hours
- Critical bugs prevent order placement or payment flow
- Backend services are down during testing windows

Resume testing if:

- Issues are resolved and verified
- Backup services or mocks are restored
- Test environment is stabilized

13. Test Case Management & Traceability

RTM ensures mapping of SRS requirements to test cases.
Example:
- FR-001 (Customer Registration) → TC-01
-FR-009 (Payment Processing) → TC-10, TC-11
-NFR-002 (Payment Response time) → TC-Perf-02

14. Test Metrics & Reporting

Metrics collected:

- Test case execution percentage
- Passed vs failed counts
- Defect density by module
- Defect aging trends
- Requirement coverage index

Reports:

- Daily test execution reports
- Weekly defect review meetings
- Final test summary report for stakeholders

15. Approvals

Role	Name	Signature / Date
QA Lead	Darshan Gowda BM	14/9/2025
Dev Lead	Jahnavi	14/9/2025
Product Owner	Isha	14/9/2025