

Software Test Plan (STP)

Project: Micro-Donation Platform

Version: 1.0

Authors: Mrunal Manjunath Kudtarkar, Meghana Saisri Bisa, Mitha M K, N.Rakshitha

Date: 16-09-2025

Status: Sample / Draft

1. Introduction

Purpose:

This document defines the test plan for the Micro-Donation Platform v1.0. It outlines the objectives, scope, strategy, resources, schedule, and responsibilities associated with testing the platform's core functionalities and integrations.

Scope:

Testing covers features such as user registration and authentication, donation initiation, transaction processing via the in-house payment gateway, security validations, transaction logging, and audit interactions. Internal bank server logic and replica synchronization mechanisms are excluded from this scope.

2. Test Items

- **ATM Authentication Module** – Validating card and PIN.
- **Cash Withdrawal Module** – Ensuring correct balance checks and dispensing.
- **Balance Inquiry Module** – Accurate retrieval and display of account balance.
- **Deposit Module** – Capturing and updating deposits (cash/cheque).
- **Fund Transfer Module** – Validating source, destination, limits, and transaction updates.
- **Admin/Maintenance Interface** – Admin login, reports, and configuration.

3. Features to be Tested

Mapped from SRS functional and non-functional requirements:

1. User Authentication & Registration

- Email/phone/social login (OAuth).
- OTP and password authentication.
- Role-based access (Donor, NGO, Admin).

2. Donation Management

- One-time and recurring donations.
- Minimum donation ₹1.
- Instant receipt and confirmation.
- Donation history and impact reports.

3. NGO / Campaign Management

- NGO registration with KYC.
- Campaign creation and approval workflow.
- Campaign progress bar updates.

4. Payment & Wallet

- Integration with multiple payment gateways (UPI, cards, wallets).
- Refund handling (failed/duplicate transactions).
- Wallet for micro-balances and round-up donations.

5. Transparency & Reporting

- Real-time NGO dashboards.
- Annual donation compliance reports (e.g., 80G).
- Social media campaign sharing.

6. Non-Functional Features

- Performance: Donation flow $\leq 3s$ for 95% of cases.
- Reliability: 99.9% uptime; replica sync $\leq 30s$.
- Security: TLS 1.2+, PCI-DSS compliance, MFA for admins.
- Usability: WCAG 2.1 AA accessibility compliance.
- Fault tolerance: Replica takeover on failure.

4. Features Not to be Tested

Third-Party Payment Gateway APIs

- No need to test external APIs like Stripe or Razorpay.
- Focus shifts entirely to your own gateway's logic and endpoints.

Bank Server Internal Logic

- Testing of the responses the gateway will receive, but not the internal workings of the bank's systems.
- Bank-side validations, fraud checks, and ledger updates are out of scope.

Low-Level Database Optimization

- Indexing, query tuning, and storage engine performance are excluded from functional testing.
- These are handled separately under performance or DBA audits.

Styling and Visual Design

- Font choices, button aesthetics, and layout polish are not tested unless they impact usability or accessibility.
- Focus remains on functional correctness and responsiveness.

Load Testing for Informational Pages

- Pages like FAQs, About Us, or static NGO profiles may be excluded from stress/load testing.

5. Test Approach / Strategy

Levels:

- Unit Tests – Validate individual modules such as donation logic, user authentication, and payment processing.
- Integration Tests – Verify interactions between platform server, payment gateway, and bank server.
- System Tests – End-to-end validation of donation workflows, user roles, and transaction flows.

Types:

- Functional Testing – Validate core features like donation initiation, transaction logging, and user management.
- Regression Testing – Ensure new updates don't break existing functionality.
- Performance Testing – Assess response time and throughput under load, especially for donation and payment flows.
- Usability Testing – Evaluate clarity, accessibility, and intuitiveness of the user interface for donors, NGOs, and admins.

Entry Criteria:

- Stable build delivered to QA
- Test data prepared and validated
- Test environment configured and accessible

Exit Criteria:

- 100% of planned test cases executed
- No open critical or high-severity defects
- All acceptance criteria met and signed off

5.1 Security Validation

- Validate user credential handling (masking passwords, no logging of sensitive data)
- TLS 1.2+ verification for all client-server and server-to-server communications
- PCI-DSS compliance checks for donation and payment data handling
- Fuzz testing for input fields (donation amount, user details, payment metadata)
- Penetration testing of authentication, donation initiation, and admin access flows

6. Test Environment

Hardware:

- ATM terminal with card reader, cash dispenser, receipt printer, deposit acceptor.

Software:

- ATM Application v1.0
- Core Banking API (sandbox mode)

Tools:

- Selenium → UI Automation
- Postman → API Testing
- JMeter → Load & Performance Testing
- Jira → Defect Tracking

Test Data:

- Dummy accounts with varying balances (low, high, zero)
- Test cards (valid, expired, blocked, stolen)
- Predefined transactions for regression

7. Test Schedule

Milestone	Date (2025)
Test case design	05-Sep-2025
Environment setup	07-Sep-2025
Test execution start	08-Sep-2025
Test execution end	20-Sep-2025
UAT (User Acceptance Testing)	22-25 Sep-2025

8. Test Deliverables

- Test Plan (this document)
- Test Cases (manual & automated)
- Test Scripts
- Test Data
- Test Execution Logs
- Defect Reports
- Test Summary Report
- Test Review Checklist
- Backup & Recovery Validation Logs

9. Roles and Responsibilities

Role	Name	Responsibility
Client Interaction & UI Tester	Mrunal	Validate client-server request/response flows, test user registration, authentication, and donation initiation, ensure usability and accessibility across supported browsers, simulate donor, NGO, and admin roles for functional coverage
Platform Server & Replica Validator	Rakshitha	Test transaction routing logic on the Micro-Donation Platform Server, validate passive updates and fault tolerance of the Replica, ensure database consistency between server and replica, simulate server failure and recovery scenarios
Payment Gateway & Bank Server Integrator	Meghana	Test transaction request handling and routing to the bank server, validate success/failure responses and error handling, ensure secure communication and compliance with financial protocols, test edge cases like timeouts, retries, and invalid payloads
Security & Validation Analyst	Mitha	Verify input validations, encryption, and secure data handling, test role-based access control (Donor, Admin, Auditor), validate audit logging and transaction traceability,

		simulate security breaches and confirm system resilienc
--	--	---

10. Risks and Mitigation

Risk	Mitigation Strategy
Delay in stable build delivery	Request early smoke builds from developers.
Test environment downtime	Maintain a backup cloud-based environment.
Dependency on ATM hardware vendor	Engage vendor early, use stubs/mock drivers for testing.
Security vulnerabilities	Conduct penetration and compliance testing early.
Limited test data	Coordinate with bank IT to generate diverse test accounts.

11. Assumptions & Dependencies

- 1. Core banking API sandbox will be stable and accessible during testing.
- 2. Valid test data (accounts, cards, balances) provided before execution.
- 3. All necessary hardware drivers (ATM/card reader equivalent stubs) available from vendor.
- 4. Payment gateway test environment available with retry/timeout simulation.
- 5. Replica server environment configured for failover and sync validation.
- 6. OAuth, SMS/email gateway, and social media APIs available for integration testing.

12. Suspension & Resumption Criteria

Suspension Criteria

Testing will be suspended if any of the following occur:

- Critical defects in the payment gateway that block transaction flow.
- Failure of bank server integration or unresponsive endpoints.

- Incomplete or corrupted test data affecting core donation scenarios.
- Unavailability of the replica server for fault tolerance validation.
- Major deviation from functional requirements in the platform server logic.
- Environment instability (e.g., database crashes, network outages).

Resumption Criteria

Testing will resume once:

- Blocking defects are resolved and verified through re-testing.
- Bank server and payment gateway endpoints are stable and responsive.
- Valid test data is restored and verified.
- Replica server is operational and syncing correctly.
- Functional deviations are addressed and aligned with updated requirements.
- Test environment is stable and meets baseline configuration.

13. Test Case Management & Traceability

RTM ensures mapping of SRS requirements to corresponding test cases for complete coverage and validation.

Example:

- MDP-F-001 (User Authentication) → TC-Auth-01, TC-Auth-02
- MDP-F-005 (Initiate Donation) → TC-Donate-01, TC-Donate-02
- MDP-F-010 (Transaction Logging) → TC-Log-01, TC-Log-02
- MDP-NF-001 (Response Time for Donation Flow) → TC-Perf-01
- MDP-NF-005 (Security Compliance) → TC-Sec-01, TC-Sec-02

14. Test Metrics & Reporting

Metrics to be Collected:

- % Test cases executed vs planned
- % Test cases passed / failed
- Defect density (defects per module / KLOC)
- Defect aging (time open before resolution)
- Requirement coverage (requirements mapped to test cases)

Reports Generated:

- Daily status reports (execution, blockers)
- Defect summary reports (critical/major/minor)
- Final Test Summary Report (execution coverage, pass rate, defect trend)

15. Approvals

Role	Name	Signature / Date
Client Interaction and UI Tester	Mrunal Manjunath Kudtarkar	<u>Mrunal</u> 19/09/2025
Platform Server and Replication Validator	N. Rakshitha	<u>Rakshitha</u> 19/09/2025
Payment Gateway and Bank	Meghana Saisri Bisa	<u>Meghana</u> 19/09/2025
Security and Validation Analyst	Mitha M K	<u>Mitha</u> 19/09/2025