# Software Requirements Specification (SRS)

**Project:** Micro Donation Platform

**Version:** 1.0

**Authors:** Prof. Animesh Giri / Meghana Bisa, Mitha M K, N Rakshitha, Mrunal Manjunath Kudtarkar

**Date:** 31-08-2025

**Status:** SRS document submitted — Awaiting Review

Submission Link: https://forms.gle/wS6VFgzuFktDDQzd8

## Revision history

| Version | Date | Author | Change summary | Approval |
|---------|------|--------|----------------|----------|
| 1.0 | 31-08-2025 | Instructor | SRS with diagrams embedded | |

## Approvals

| Role | Name | Signature / Email | Date |
|------|------|-------------------|------|
| Course Coordinator | Prof. Animesh Giri | Animeshgiri@pes.edu | |

# Table of Contents

# 1. Introduction

### 1.1 Purpose

This document is a Software Requirements Specification (SRS) for a Micro Donation Platform. It defines the functional and non-functional requirements, system architecture, interfaces, and verification criteria. The platform is designed to facilitate small-scale donations from users to charitable organizations via a secure and scalable digital infrastructure. This document serves as a reference for developers, testers, integrators, and evaluators involved in the system's design, implementation, and assessment.

### 1.2 Scope

Covers user interactions through the micro-donation platform's web and mobile interfaces, integration with payment gateways and donor management systems for transaction processing and fund allocation, platform maintenance functions, and operational monitoring.
Excludes internal financial institution operations beyond exposed APIs, third-party fundraising activities, and external donor engagement channels not managed by the platform.

### 1.3 Audience

Developers, QA Engineers, System Integrators, Maintenance Technicians, and Assessment Evaluators.

### 1.4 Definitions List of acronyms:

- UI – User Interface
- API – Application Programming Interface
- OTP – One-Time Password
- TLS – Transport Layer Security
- RTO – Recovery Time Objective
- RPO – Recovery Point Objective
- DB – Database
- PG – Payment Gateway
- Replica Server – Backup server mirroring the primary server

- Client – End-user application for donation interaction
- Bank Server – External system for account verification and transaction settlement
- MDP-Micro donation platform

# 2. Overall description

**2.1 Product perspective**

The Micro-Donation Platform is a distributed client-server system that connects donors, a central server, and financial institutions(bank server, payment gateway).It enables micro-level donations to verified causes with minimal transaction costs.

The architecture consists of:

- Client Application (web/mobile) for users to register, authenticate, and donate.
- Central Server managing donations, maintaining donor and ensuring synchronization with a replica server for redundancy
- Bank Server Integration for account verification, fund transfers, and settlement.
- Payment Gateway for secure processing of small transactions.
- Databases:
  - One for server & replica (donation records, donor profiles).
  - One for bank server (account balance, transaction validation).

**2.2 Major Product Functions**

- User registration and authentication.
- Browsing campaigns/causes
- Micro-donation transaction initiation.
- Secure communication with payment gateway and bank server.
- Transaction confirmation and receipt generation.
- Synchronization between server and replica.
- Administrative monitoring and reporting.
- Real-time donation status and history for users.
- Fraud detection and duplicate transaction prevention.
- Database backup and recovery support.

**2.3 User Roles and Characteristics**

- Donors: Basic digital literacy, mobile/online banking familiarity.
- NGOs/Beneficiaries: Need transparency and access to reports.

- System Administrators: Manage campaigns, monitor transactions, ensure replica sync.
- Bank Operators: Oversee settlements and compliance.

**2.4 Operating Environment**

- Web clients accessed via modern desktop browsers (Chrome, Firefox, Edge).
- Local development Server & replica.
- Secure internet connectivity.
- Database servers with backup/recovery.

**2.5 Constraints**

- Must comply with **RBI guidelines** for payment systems.
- PCI-DSS compliance for cardholder data.
- High availability requirement (99.9% uptime).
- Low latency (<5 seconds per transaction).
- Cost-effective transactions suitable for micro-donations.

# 3. External interface requirements

3.1 **Client-Side Features**

These are the features accessible to end users (donors, NGOs, admins):

- User Registration & Authentication
  - Secure login and sign-up
  - Role-based access (Donor, NGO, Admin)
- Initiate Donation
  - Select donation amount and recipient
  - Trigger transaction request to the server
- Transaction Status Feedback
  - Real-time success/failure notifications
  - Confirmation receipts or logs
- Receive Donations
  - NGOs or recipients get notified of incoming funds
  - Option to view donation history
- Audit & Admin Controls
  - Admin can validate transactions
  - Auditor can access logs for compliance

## 3.2 **Payment Gateway Integration**

Handles actual financial transactions:

- Transaction Request Handling
  - Accepts payment initiation from platform server
  - Communicates securely with bank server
- Security Validations
  - Fraud detection, encryption, and compliance checks
  - Ensures PCI-DSS or similar standards

## 3.3 **Bank Server Interaction**

Finalizes the transaction:

- Transaction Processing
  - Verifies account details and fund availability
  - Sends success/failure response to platform
- Database Logging
  - Stores transaction records for audit and rollback

## 3.4 **Replica Server (Passive)**

Supports fault tolerance:

- Passive Updates
  - Mirrors platform server data for recovery
  - Ensures high availability and reliability

# 4. Detailed System Features (Functional Requirements)

## 4.1 **User Authentication & Registration**

Description: The platform shall authenticate donors, NGOs, and administrators securely and allow account creation with proper role-based access.

| Req ID | Requirement (shall...) | Type | Priority | Source/Stakeholder | Acceptance criteria / Test case ref | Comments / Dependencies |
|--------|------------------------|------|----------|--------------------|-----------------------------------|-------------------------|
| MDP-F-001 | The system shall allow users to register using email, phone, or social login (Google/FB). | Functional | High | Donor/NGO | AC-MDP-F-001: Registration succeeds with valid inputs. Test: TC-Auth-01 | Depends on OAuth APIs |
| MDP-F-002 | The system shall authenticate users via password or OTP. | Functional | High | Security | AC-MDP-F-002: Login succeeds with valid credentials. Test: TC-Auth-02 | Requires SMS/Email OTP gateway |
| MDP-F-003 | The system shall provide role-based access (Donor, NGO Admin, Platform Admin). | Functional | High | Compliance | AC-MDP-F-003: Role permissions enforced. Test: TC-Auth-03 | RBAC framework |

## 4.2 Donation Management

Description: The platform shall support one-time and recurring micro-donations, track history, and provide confirmations.

| Req ID | Requirement (shall...) | Type | Priority | Source/ Stakeholder | Acceptance criteria / Test case ref | Comments / Dependencies |
|---|---|---|---|---|---|---|
| MDP-F-010 | The system shall allow donors to make micro-donations starting from ₹1. | Functional | High | Donor | AC-MDP-F-010: Donation succeeds with valid payment. Test: TC-Donate-01 | Payment Gateway API |
| MDP-F-011 | The system shall support one-time and recurring donations. | Functional | Medium | Donor | AC-MDP-F-011: Recurring donation schedule executed. Test: TC-Donate-02 | Cron/Subscription service |
| MDP-F-012 | The system shall generate instant receipts and confirmation notifications. | Functional | High | Donor/NGO | AC-MDP-F-012: Receipt sent via email/SMS. Test: TC-Donate-03 | Email/SMS integration |
| MDP-F-013 | The system shall allow donors to view donation history and impact reports. | Functional | Medium | Donor | AC-MDP-F-013: Past donations listed correctly. Test: TC-Donate-04 | Data audit |

## 4.3 NGO / Campaign Management

Description: The platform shall enable NGOs to register, undergo verification, and create/manage campaigns.

| Req ID | Requirement (shall...) | Type | Priority | Source/Stakeholder | Acceptance criteria / Test case ref | Comments / Dependencies |
|---|---|---|---|---|---|---|
| MDP-F-020 | The system shall allow NGOs to register and undergo verification (KYC). | Functional | High | Platform Admin | AC-MDP-F-020: NGO listed only after approval. Test: TC-NGO-01 | KYC API |
| MDP-F-021 | The system shall allow NGOs to create and manage donation campaigns. | Functional | High | NGO | AC-MDP-F-021: Campaign visible after approval. Test: TC-NGO-02 | Requires moderation |
| MDP-F-022 | The system shall show campaign progress bars (% funded). | Functional | Medium | Donor | AC-MDP-F-022: Progress updated after donation. Test: TC-NGO-03 | DB update logic |

## 4.4 Payment & Wallet

Description: The platform shall provide secure payment integrations and a wallet feature for micro-balances.

| Req ID | Requirement (shall...) | Type | Priority | Source/ Stakeholder | Acceptance criteria / Test case ref | Comments / Dependencies |
|---|---|---|---|---|---|---|
| MDP-F-030 | The system shall integrate with multiple payment gateways (UPI, Cards, Wallets). | Functional | High | Donor | AC-MDP-F-030: Payment succeeds with valid credentials. Test: TC-Pay-01 | Requires PG API |
| MDP-F-031 | The system shall support refunds for failed/duplicate transactions. | Functional | Medium | Donor/Admin | AC-MDP-F-031: Refund processed. Test: TC-Pay-02 | Payment Gateway support |
| MDP-F-032 | The system shall maintain a secure wallet for small balances and round-up donations. | Functional | Medium | Donor | AC-MDP-F-032: Balance updated correctly. Test: TC-Pay-03 | Virtual wallet |

## 4.5 Transparency & Reporting

Description: The platform shall provide dashboards, reports, and sharing features for donors and NGOs.

| Req ID | Requirement (shall...) | Type | Priority | Source/ Stakeholder | Acceptance criteria / Test case ref | Comments / Dependencies |
|---|---|---|---|---|---|---|
| MDP-F-040 | The system shall provide real-time dashboards for NGOs to track donations. | Functional | High | NGO | AC-MDP-F-040: Dashboard updated within 5s. Test: TC-Report-01 | Analytics service |
| MDP-F-041 | The system shall provide annual donation reports for compliance (e.g., 80G receipts). | Functional | High | NGO/Donor | AC-MDP-F-041: Downloadable PDF generated. Test: TC-Report-02 | Depends on reporting module |
| MDP-F-042 | The system shall allow donors to share campaigns on social media. | Functional | Medium | Donor | AC-MDP-F-042: Campaign link shared successfully. Test: TC-Report-03 | Social Media API |

# 5. Non-functional requirements (detailed)

NFRs below are measurable and tied to test plans. IDs MDP-NF-###.

| Req ID | Requirement | Category | Priority | Acceptance criteria / Measurement |
|---|---|---|---|---|
| MDP-NF-001 | End-to-end donation transaction (from client initiation to confirmation) shall complete ≤ 3 seconds for 95% of cases under normal load. | Performance | High | 95th percentile ≤ 3s in staging tests. Test: TC-Perf-01 |
| MDP-NF-002 | Platform shall maintain 99.9% availability monthly, excluding scheduled maintenance. | Reliability | High | Uptime logs ≥99.9% monthly. Test: Ops-Metrics-01 |
| MDP-NF-003 | All sensitive user and payment data must comply with PCI-DSS and not be stored in plaintext. | Security/Compliance | High | PCI-DSS audit checklist pass. Test: TC-Sec-01 |
| MDP-NF-004 | System shall retain timestamped donation logs for 5 years for audit and transparency | Audit/Data Retention | High | Verified archival and retrieval. Test: TC-Log-01 |
| MDP-NF-005 | Platform shall support accessibility features (keyboard navigation, screen reader, contrast modes) conforming to WCAG 2.1 AA. | Usability/Accessibility | Medium | Accessibility audit pass. Test: TC-UX-01 |
| MDP-NF-006 | Replica server shall sync with primary server within 30 seconds of any transaction or update. | Fault Tolerance | High | Sync logs show ≤30s lag. Test: TC-Replica-01 |

| MDP-NF-007 | Payment gateway integration shall support retry logic for transient failures with max 3 attempts. | Resilience | Medium | Retry logic verified in test cases. Test: TC-PG-02 |
|---|---|---|---|---|

## 5.1. Security

### 5.1.1 Security Objectives

- **Ensure Secure Data Transmission**

  All communication between clients, servers, and external services (e.g., payment gateway, bank server) must be encrypted using industry-standard protocols (TLS 1.2 or higher) to prevent data interception or tampering.

- **Protect Donor and Charity Information**

  Sensitive user data—including personal details, donation history, and payment credentials—must be securely stored and handled in compliance with data protection standards.

- **Prevent Unauthorized Access**

  Implement robust authentication and authorization mechanisms for all system components, including multi-factor authentication (MFA) for administrative access and role-based controls for charity management.

- **Enable Auditability and Traceability**

  Maintain detailed, timestamped logs of all financial transactions, administrative actions, and system events to support transparency, fraud detection, and regulatory audits.

### 5.1.2 Security Requirements

| Req ID | Requirement (shall...) | Type | Priority | Acceptance criteria / Test case ref |
|---|---|---|---|---|
| MDP-SR-001 | All data in transit between client, server, and payment gateway shall be encrypted using TLS 1.2+ | Data Transmission | High | Verified via network inspection. Test: TC-Sec-02 |
| MDP-SR-002 | User authentication shall include OTP verification for donation transactions above ₹500. | Authentication | Medium | OTP flow validated. Test: TC-Auth-01 |
| MDP-SR-003 | Admin access to server and replica shall require multi-factor authentication (MFA). | Access Control | High | MFA setup and test pass. Test: TC-Admin-01 |
| MDP-SR-004 | Payment gateway shall not store CVV or PAN in any form. | Compliance | High | Gateway audit pass. Test: TC-PG-01 |

# 6. Quality attribute

- Performance:
    - 90% of donation transactions should complete within **≤5 seconds**.
- Reliability:
    - Server + Replica should ensure **99.9% uptime** monthly.
- Security:
    - End-to-end encryption using **TLS 1.2+**, no sensitive data stored in plaintext.
- Scalability:
    - System should support **10,000 concurrent users** and **100,000 daily transactions**.
- Usability:
    - Simple, intuitive UI support for accessibility.
- Auditability:
    - Transaction logs stored for minimum 7 years.
- Maintainability:
    - Modular design with separate APIs for donation, reporting, and monitoring

**Acceptance tests:**
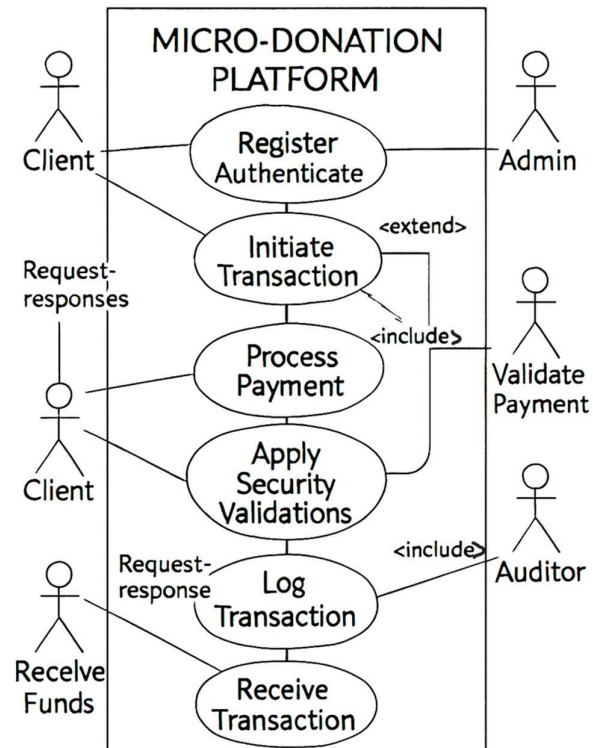
- Authentication Tests:
    - AT-001: Valid login with correct credentials grants access.
    - AT-002: Invalid login shows error after 3 failed attempts.
- Donation Tests:
    - AT-010: Donation of valid amount (₹10–₹100) succeeds, balance deducted, confirmation generated.
    - AT-011: Donation exceeding limit rejected.
    - AT-012: Transaction recorded in both server and replica.
- Payment Gateway Tests:
    - AT-020: Secure connection (TLS) established for all transactions.
    - AT-021: Failed bank server connection triggers retry with backoff.
- Performance Tests:
    - AT-030: 90% of transactions complete in ≤5 seconds under normal load.
    - AT-031: System handles 10,000 concurrent donor requests without crash.
- Security Tests:
    - AT-040: No sensitive card/account data stored in plaintext.
    - AT-041: Logs contain masked details (e.g., last 4 digits only).
- Reliability Tests:
    - AT-050: If server fails, replica takes over within ≤30 seconds.
    - AT-051: Database backup successfully restored in disaster recovery test.

# 7. System models and diagrams

**7.1 UML Use-Case diagram (Basic UML Diagram)**
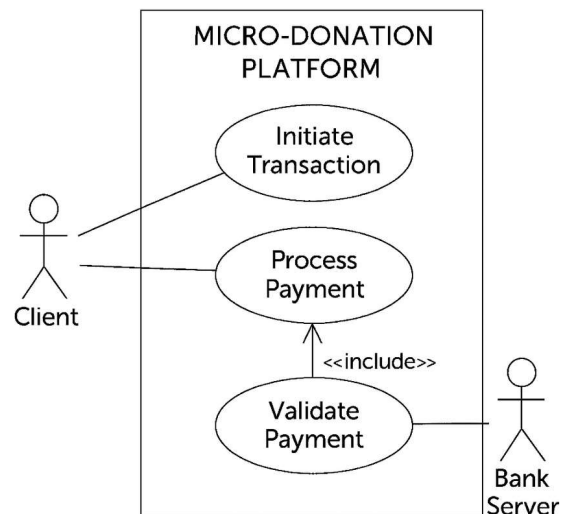


**7.2 UML Use-Case diagram (Detailed UML Diagram)**

# 8. Requirements Traceability Matrix (RTM)

| Req ID | Requirement short | Section ref / Design Spec | Module | Test case(s) | Status (N/P/A) | Comments |
|---|---|---|---|---|---|---|
| MDP-F-001 | User Registration | 4.1 / DS-Auth-01 | AuthModule | TC-Auth-01 | N | Needs OAuth |
| MDP-F-002 | User Authentication | 4.1 / DS-Auth-02 | AuthModule | TC-Auth-02 | N | OTP Gateway |
| MDP-F-003 | Role-based Access | 4.1 / DS-Auth-03 | AuthModule | TC-Auth-03 | N | RBAC |
| MDP-F-010 | Micro Donation | 4.2 / DS-Donate-01 | DonationModule | TC-Donate-01 | N | PG integration |
| MDP-F-011 | Recurring Donation | 4.2 / DS-Donate-02 | DonationModule | TC-Donate-02 | N | Scheduler |
| MDP-F-012 | Instant Receipts | 4.2 / DS-Donate-03 | NotificationModule | TC-Donate-03 | N | Email/SMS |
| MDP-F-013 | Donation History | 4.2 / DS-Donate-04 | DonationModule | TC-Donate-04 | N | DB logs |
| MDP-F-020 | NGO Verification | 4.3 / DS-NGO-01 | NGOModule | TC-NGO-01 | N | KYC API |
| MDP-F-021 | Campaign Management | 4.3 / DS-NGO-02 | NGOModule | TC-NGO-02 | N | Approval workflow |
| MDP-F-022 | Campaign Progress | 4.3 / DS-NGO-03 | NGOModule | TC-NGO-03 | N | DB updates |

| MDP-F-030 | Payment Gateway | 4.4 / DS-Pay-01 | PaymentModule | TC-Pay-01 | N | Multiple PGs |
|---|---|---|---|---|---|---|
| MDP-F-031 | Refunds | 4.4 / DS-Pay-02 | PaymentModule | TC-Pay-02 | N | PG support |
| MDP-F-032 | Wallet Balance | 4.4 / DS-Pay-03 | WalletModule | TC-Pay-03 | N | Round-up logic |
| MDP-F-040 | NGO Dashboard | 4.5 / DS-Report-01 | AnalyticsModule | TC-Report-01 | N | Real-time sync |
| MDP-F-041 | Annual Reports | 4.5 / DS-Report-02 | ReportingModule | TC-Report-02 | N | Compliance |
| MDP-F-042 | Social Sharing | 4.5 / DS-Report-03 | SharingModule | TC-Report-03 | N | Social API |