

# Product Requirements Document (PRD)

## Feature: Payment Retry & Failure Handling for UPI Transactions

Author: Vedakshi Gawde | Role: Business Analyst / Product Analyst

### Background & Context

UPI payment failures are common due to network issues, bank timeouts, or validation errors. Unclear outcomes lead to repeated retries, customer anxiety, and increased support tickets.

### Problem Statement

Users lack clarity on whether to retry or wait after a failed or pending UPI transaction, leading to duplicate attempts and reduced trust.

### Goals & Success Metrics

- Reduce payment-related complaints by 15%.
- Improve transaction success rate by 10%.
- Reduce duplicate retry attempts.

### Scope In Scope:

Failure detection, retry logic, user messaging, refund visibility.

### Out of Scope:

- Bank-side logic changes
- new payment modes.

### User Journey

User initiates payment → transaction fails → system classifies failure → clear retry or wait guidance shown → refund or success confirmation displayed.

### Functional Requirements

- FR-01: Classify failures.
- FR-02: Controlled retries.
- FR-03: Contextual user messaging.
- FR-04: Refund and reversal visibility.

### Non-Functional Requirements

- Prevent duplicate transactions.
- Maintain performance.
- Ensure audit logging.

Edge Cases Retry during pending state, delayed bank response, app closure during retry.

**Dependencies & Assumptions** Dependent on bank response codes and UPI network availability. **Risks & Mitigation** Duplicate debit risk mitigated through strict retry validation.

## Open Questions

- Maximum retry count?
- Refund SLA to display?
- Notification triggers?