Feature Design PRD: Smart Error Recovery

> **Author:** Abhineet Jain
> **Version:** 2.0 | **Last Updated:** January 19, 2026
> **Status:** Ready for Review | **Live Demo:** pocketful.vercel.app

---

1. Overview

1.1 Feature Summary

The **Smart Error Recovery** system transforms generic, unhelpful error messages into intelligent, context-aware recovery experiences. Instead of leaving users stuck with “Something went wrong,” the system classifies errors, explains them in plain language, and offers actionable next steps.

1.2 Problem Statement

Current State:
When Pocketful encounters API failures, data unavailability, or network issues, users see a generic message:

> *“Something went wrong! Internal Server Error”*

This message:

- Provides no explanation of what went wrong
- Offers no alternative actions
- Fails to distinguish user-fixable issues (network) from platform problems (server outage)
- Erodes trust during critical trading moments

User Impact:

- Trust erosion during high-stakes moments
- High session abandonment after errors
- Significant support ticket volume for error-related queries

- Users unsure whether to retry, wait, or abandon the action

1.3 Target Personas

Persona	Profile	Pain Point	Value from SER
Priya, 28	Active F&O trader, 10+ trades/week	Loses money when errors delay action	Contextual alternatives keep her trading
Raj, 24	First-time investor, confused by jargon	"Internal Server Error" is scary	Plain-English explanations reduce anxiety
Amit, 45	Tier-2 city, patchy network	Doesn't know if it's his network or app	Network detection + RM access provides support

---

2. Value Proposition

> *"Transform every error into an opportunity to demonstrate platform reliability."*

Why This Feature Matters

Business Impact:

- **Trust Recovery:** Errors are inevitable; recovery experience determines if users stay or leave
- **Differentiation:** Neither Zerodha (sparse handling) nor Groww (friendly but generic) offers personalized recovery
- **Support Reduction:** Clear guidance reduces ticket volume by ~40% (industry benchmark)

User Impact:

- Users feel supported, not abandoned
- Reduces decision paralysis during critical trading moments
- Builds long-term platform loyalty through transparency

How It Works

The Smart Error Recovery System:

1. **Diagnoses** the root cause and translates it into user-friendly language
2. **Offers** contextually relevant alternatives based on user intent
3. **Integrates** with RM support for complex issues (Pocketful's unique advantage)
4. **Learns** from recovery patterns to personalize future error handling

---

3. Success Criteria

3.1 Primary Metrics

Metric	What We're Measuring	How to Validate
Error Recovery Rate	% of users who complete intended action after encountering an error	Analytics: Track user actions post-error (retry success, alternative selection, session continuation)
Session Continuation	% of sessions that continue after error vs. abandon	Compare session drop-off rates before/after launch
Support Ticket Volume	Error-related support queries	Track tickets tagged with error-related categories

3.2 Qualitative Signals

- User feedback mentions "helpful error messages"
- Reduced negative app store reviews citing reliability
- Internal QA reports fewer "dead end" scenarios

---

4. User Stories

4.1 Primary Stories

ID	User Story	Priority	Acceptance Criteria
SER-001	As a trader, I want to understand WHY an error occurred so I can decide	P0	Error message includes category (Network/Data/Server) and plain-English explanation.

ID	User Story	Priority	Acceptance Criteria
	whether to retry or try a different approach.		
SER-002	As a user experiencing an error, I want to see alternative actions so I'm not stuck.	P0	At least 2 contextual alternatives displayed based on error type and intended action.
SER-003	As a user on a poor network, I want to know if the error is on my end so I can troubleshoot my connection.	P1	Client-side network check runs before displaying error; shows network-specific guidance if detected.
SER-004	As a trader during market hours, I want the recovery options to appear instantly.	P1	During 9:15 AM - 3:30 PM IST, system loads <100ms with pre-computed responses.

4.2 Edge Cases

ID	Scenario	Handling
SER-E01	Multiple errors in rapid succession (>3 in 60 seconds)	Show consolidated error summary instead of multiple toasts
SER-E02	Error occurs when user is offline	Show cached portfolio with "Last updated" timestamp; queue action for when online
SER-E03	Accessibility requirement	All elements have proper labels for screen readers (WCAG 2.1 AA)

---

5. User Flow

5.1 Happy Path: Error Recovery Journey

```
 [1. USER ACTION]              [2. ERROR DETECTED]
┌──────────────────┐         ┌──────────────────────┐
```

```
│ User taps        │─────────│ API returns error    │
│ "Technicals"     │         │ or timeout           │
└──────────────────┘         └──────────┬───────────┘
                                        │
                                        │
                                        ▼
                              [3. RECOVERY ACTIVE]
                             ┌────────────────────────────────┐
                             │ ⚠️ Data Temporarily Unavailable │
                             │                                │
                             │ Technical indicators for       │
                             │ IRCTC are being updated.       │
                             │ Usually takes ~30 seconds.     │
                             │                                │
                             │ ┌─ WHAT YOU CAN DO ──────────┐ │
                             │ │ 🔄 Retry in 30 seconds     │ │
                             │ │ 📊 View Similar: RVNL      │ │
                             │ │ 📈 Use Last Known Data     │ │
                             │ └────────────────────────────┘ │
                             │                                │
                             │ 💬 Report this issue           │
                             └────────────────────────────────┘
                                        │
                  ┌─────────────────────┼──────────────────────┐
                  ▼                     ▼                      ▼
          [4a. RETRY SUCCESS]   [4b. ALTERNATIVE SELECTED] [4c. CACHED DATA]
         ┌──────────────────┐  ┌──────────────────────┐  ┌─────────────────
         │ ✅ Data loaded   │  │ User views RVNL      │  │ User sees data
         │                  │  │ technicals instead   │  │ with timestamp
         └──────────────────┘  └──────────────────────┘  └─────────────────
                  │                     │                      │
                  └─────────────────────┼──────────────────────┘
                                        │
                                        ▼
                              [5. RECOVERY LOGGED]
                             ┌────────────────────────────────┐
                             │ Successful path saved for      │
                             │ future personalization         │
                             └────────────────────────────────┘
```

5.2 Cascading Error Handling

When multiple errors occur within a short timeframe:

```
 Error 1 → Message shown
     │
     ▼ (within 60 seconds)
Error 2 → Message updated silently
     │
     ▼ (within 60 seconds)
Error 3+ → CONSOLIDATED VIEW

┌────────────────────────────────────┐
│ ⚠️ Multiple Issues Detected         │
│                                    │
│ We're experiencing temporary       │
│ connectivity issues.               │
│ 3 requests affected.               │
│                                    │
│ [🔄 Retry All]  [⏸️ Wait & Retry]   │
└────────────────────────────────────┘
```

---

6. Wireframe Components

6.1 Component Breakdown

Component	Purpose	States
Error Header	Displays categorized error type with icon	Warning (⚠️), Critical (🚫), Info (ℹ️)
Actionable Toast	Plain-language root cause + estimated resolution	Loading, Loaded
Recovery Actions	2-4 contextual buttons	Default, Pressed, Disabled
Progress Indicator	Shows retry countdown or loading	Counting, Loading, Complete

Component	Purpose	States
Feedback Link	Opens lightweight issue reporter	Default, Expanded

6.2 Visual Layout

```
┌─────────────────────────────────────────────┐
│ [ICON] ERROR CATEGORY            [X Close]  │  ← High contrast header
├─────────────────────────────────────────────┤
│                                             │
│  Explanation text in clear language         │  ← Primary content
│  with estimated wait time if applicable.    │
│                                             │
│  ┌───────────────────────────────────────┐  │
│  │ [Primary Action Button]               │  │  ← Main CTA
│  └───────────────────────────────────────┘  │
│                                             │
│  [Secondary Action]  [Secondary Action]     │  ← Alternatives
│                                             │
│  ─────────────────────────────────────      │
│  💬 Report this issue                       │  ← De-emphasized
└─────────────────────────────────────────────┘
```

7. Error Classification System

The system requires a backend error taxonomy service to classify errors. Here's the proposed classification:

Error Type	User Message Example	Recovery Options
`DATA_UNAVAILABLE`	"Technical data for this instrument is being refreshed"	Retry, View Similar, Use Cached
`NETWORK_TIMEOUT`	"Your connection seems slow. Check your network."	Retry, Switch to Lite Mode

Error Type	User Message Example	Recovery Options
`SERVER_ERROR`	"We're experiencing high load. Our team is aware."	Retry Later, View Status Page
`RATE_LIMITED`	"Too many requests. Please wait a moment."	Auto-retry countdown

---

8. Technical Considerations

8.1 Key Requirements

Requirement	Description
Latency	UI must render within 100ms (pre-fetch alternatives)
Offline Support	Basic recovery should work even when partially offline
Client Diagnostics	Should detect if user's network is the issue
Error Logging	All errors and recovery paths logged for analytics

8.2 Dependencies

Dependency	Owner	Status
Error classification service	Backend Team	Needs development
Similar instruments mapping	Data Team	Static fallback available
Client network diagnostics	Mobile Team	Standard SDK available

8.3 Security & Compliance

- Error messages must NOT expose internal system details or stack traces
- Recovery paths must validate user permissions before offering

- All logging must be anonymized for analytics

---

9. Rollout Plan

9.1 Phased Rollout

Phase	Scope	Criteria to Proceed
Alpha	Internal team (50 users)	No critical bugs
Beta	10% of users	Recovery rate improved vs. control
GA	100% of users	Metrics stable; support not overwhelmed

9.2 Rollback Conditions

- Recovery rate drops below current baseline for 24 hours
- Recovery UI load time exceeds 1 second at p95
- User-reported issues spike significantly

---

10. Success Validation

10.1 How We'll Know It's Working

Signal	Method
Users completing actions after error	Analytics funnel tracking
Reduced error-related tickets	Support ticket categorization
Improved sentiment	App store reviews; in-app feedback
Session continuation	Compare abandonment rates pre/post launch

10.2 User Acceptance Test Scenarios

ID	Scenario	Input / Action	Expected Result	Priority
ER-001	**Server 500 Error**	User taps "Buy" on Order Pad → Simulate 500 API Error	"Retry" action toast appears <100ms; Message: "Connection slow"	P0
ER-002	**Network Failure**	User tries to load Watchlist → Simulate Offline Mode	"You are offline" message; Cached data displayed (no empty state)	P0
ER-003	**Latency Check**	Market Hours (9:15-3:30) → Trigger Error	Render time <100ms (Measure via perf monitor)	P0
ER-004	**Rate Limit**	User rapid-taps refresh 5 times	Consolidated "Too many requests" toast; cooldown timer shown	P1
ER-005	**Unclear Error**	Trigger generic 400 error (Unknown cause)	Message blames System ("We are having trouble"), NOT User	P1

---

11. Competitive Advantage

Platform	Error Handling Approach	Our Advantage
Zerodha	Minimal, functional messages	We offer personalized recovery paths
Groww	Friendly but generic	We classify errors and offer alternatives
Upstox	Basic retry prompts	We provide context and transparency

Positioning: Pocketful becomes the platform that helps you when things go wrong, not one that leaves you stranded.

---

12. Risk Assessment

Identified Risks

Risk ID	Risk Type	Description	Probability	Impact	Mitigation
SER-R1	Technical	Error classification inaccuracy: System may misclassify errors, providing wrong context	Medium	High	Conservative classification: If ambiguous, default to “Connection Issue” (blame system), NEVER blame user.
SER-R2	UX	Alternative overload: Too many recovery options could confuse users	Low	Medium	Limit to max 2 critical alternatives; use compact “Toast” design
SER-R3	Adoption	Users ignore wizard: May dismiss without reading if too intrusive	Medium	Medium	Make dismissible but not auto-dismiss; track engagement metrics
SER-R4	Performance	Latency overhead: Classification logic delays UI	Low	High	Pre-fetch alternative maps; Render UI immediately (<50ms) before classification finishes
SER-R5	Data	Privacy concerns: Logging user actions during errors	Low	Medium	Anonymize all logged data; transparent privacy notice

Risk Response Strategy

Risk ID	Response	Owner	Trigger
SER-R1	Default to “We’re having trouble” (System blame) for all ambiguous errors	Engineering	>0% ambiguity in testing
SER-R2	A/B test 2 vs. 3 alternatives; choose based on recovery rate	Product	Pre-launch testing
SER-R3	Add “helpful” rating to wizard; iterate based on feedback	Product	<50% engagement rate
SER-R4	Pre-fetch “Alternative Instruments” mapping at app launch	Engineering	P99 latency >50ms
SER-R5	Legal review of logging; implement data retention policy	Product + Legal	Before launch

---

Appendix: RACI Matrix

Deliverable	Product (You)	Engineering	Design	QA
PRD & Requirements	**R/A**	C	C	I
Error Taxonomy Design	C	**R/A**	I	C
System UI Design	A	I	**R**	I
Implementation	C	**R/A**	I	C
Test Cases	**R**	C	I	**A**
Rollout Decision	**R/A**	C	I	C

Legend: R = Responsible, A = Accountable, C = Consulted, I = Informed

Appendix: User Flow Diagram

Smart Error Recovery Flow

```
┌──────────────────────────────────────────────────────────────────────────
│                         USER FLOW: SMART ERROR RECOVERY
└──────────────────────────────────────────────────────────────────────────


    ┌─────────┐
    │  START  │ User taps "Technicals" or triggers data request
    └────┬────┘
         │
         ▼
    ┌─────────────┐
    │  API Call   │ Request sent to data provider
    └──────┬──────┘
           │
           ▼
    ◆ Response OK? ◆──────── YES ─────► [Display Data] ──► END
           │
           NO
           │
           ▼
    ┌─────────────────────┐
    │  Error Detected     │ <50ms detection (Pre-fetched)
    │  Classify Error     │
    └──────────┬──────────┘
               │
    ┌──────────┴──────────┬────────────────────┐
    │                     │                    │
    ▼                     ▼                    ▼
┌─────────┐        ┌───────────┐        ┌──────────┐
│ NETWORK │        │   DATA    │        │  SERVER  │
│  ISSUE  │        │UNAVAILABLE│        │  ERROR   │
└────┬────┘        └─────┬─────┘        └────┬─────┘
     │                   │                   │
     ▼                   ▼                   ▼
┌──────────┐       ┌───────────┐       ┌───────────┐
```

```
   │ Check    │         │ View      │        │ Retry     │
   │ Network  │         │ Similar   │        │ Later     │
   ├──────────┤         │ Symbols   │        │           │
   │ Retry    │         ├───────────┤        │ Contact   │
   │          │         │ Use Cache │        │ Support   │
   └────┬─────┘         └─────┬─────┘        └─────┬─────┘
        │                     │                    │
        │                     │                    │
        └─────────────────────┼────────────────────┘
                              │
                              │
                              ▼
                   ┌─────────────────┐
                   │ Log Recovery    │
                   │ Path for Future │
                   │ Personalization │
                   └────────┬────────┘
                            │
                            │
                            ▼
                   ┌─────────────────┐
                   │  SESSION        │
                   │  CONTINUES      │
                   └─────────────────┘
```

Flow States Legend

Shape	Meaning
[ ] Rectangle	User Action / Process
◆ Diamond	Decision Point
( ) Rounded	Start / End
Arrows	Flow Direction

> **Document Owner:** Abhineet Jain
> **Stakeholders:** Engineering, Design, QA
> **Review Cycle:** As needed during development