

# Detailed Product Requirements Document (PRD): LedgerOne v1.0

**Project Name:** LedgerOne

**Author:** Product Lead

**Date:** February 16, 2026

**Target Market:** Ethiopia (Emerging Markets)

---

## 1. Product Vision & Strategic Objective

### 1.1 Executive Summary

LedgerOne is a decentralized personal finance manager (PFM) designed for the "Pre-Open Banking" era. In markets like Ethiopia, financial data is siloed across various banking apps (CBE Mobile, Telebirr, Awash, BOA). LedgerOne bypasses the lack of APIs by utilizing the user's notification tray and SMS inbox as a data stream.

### 1.2 The "Open Banking Gap"

Most PFM apps (Mint, YNAB) rely on Plaid or Salt Edge. These do not work in Ethiopia. LedgerOne fills this gap by turning unstructured SMS text into structured financial data locally on the user's device.

### 1.3 Strategic Goals

- **Trust First:** Achieve 100% data privacy by ensuring no transaction text ever leaves the device.
  - **Automation:** Reduce manual entry from 100% to <5%.
  - **Verification:** Solve the "SMS Spoofing" problem using bank-issued receipt URLs.
- 

## 2. Detailed User Personas & Journey Maps

### 2.1 Persona A: "The Multi-Bank Freelancer" (Abebe)

- **Behavior:** Receives USD via bank transfer (CBE), pays rent via Telebirr, buys groceries via BOA.
- **Pain Point:** Never knows his "Net Worth" across all three platforms.
- **Goal:** A single number showing total liquidity.

### 2.2 Persona B: "The Micro-Merchant" (Selam)

- **Behavior:** Accepts 30+ Telebirr payments daily in her shop.
  - **Pain Point:** Needs to verify if a customer actually paid or just showed a fake SMS.
  - **Goal:** Instant "Verified" badge on the LedgerOne dashboard.
- 

### 3. Functional Requirements (Deep Dive)

#### 3.1 Transaction Ingestion Engine (TIE)

The TIE is the core intellectual property. It consists of three modules:

##### 3.1.1 The SMS Listener (BroadcastReceiver)

- **Constraint:** Must filter by Sender ID (e.g., "CBE", "8099", "Awash").
- **Regex Engine:** A modular system where patterns are updated via a remote config (JSON) without app updates.
- **Example Regex (CBE):** `Amt: ([\d,.]+) ETB .* Ref: (\w+) .* Balance: ([\d,.]+)`

##### 3.1.2 The Notification Listener Service

- **Function:** Intercepts push notifications from banking apps.
- **Logic:** Some banks don't send SMS but do send pushes. The app must parse the `StatusBarNotification` object.

##### 3.1.3 Manual Entry & OCR (Phase 2)

- **Feature:** User takes a photo of a physical receipt or ATM slip.
- **Tech:** Google ML Kit (on-device) to extract Date, Amount, and Vendor.

#### 3.2 Receipt Verification System (RVS)

This feature differentiates LedgerOne from "SMS Trackers."

1. **URL Extraction:** Detect links within SMS (e.g., <https://receipt.cbe.com.et/v/XXXX>).
2. **Background Scraper:** A headless worker fetches the HTML content of the receipt.
3. **Data Extraction:** Extracts the official "Success" status and Transaction ID from the bank's own website.
4. **Verification Levels:**
  - **Level 0 (Manual):** User typed it.
  - **Level 1 (Detected):** Found in SMS/Notification.
  - **Level 2 (Verified):** Scraped and confirmed from the bank's official web portal.

#### 3.3 Dashboard & Analytics

- **Liquidity Card:** Sum of the "Last Known Balance" from every detected bank.
- **The "Burn Rate" Indicator:** A velocity metric showing how fast the user is spending compared to the previous 7 days.
- **Category Breakdown:** Automatic tagging based on keywords (e.g., "Yango" → Transport; "Zmall" → Food).

---

## 4. Technical Architecture

### 4.1 System Diagram (High-Level)

- **Layer 1 (Input):** Android System (SMS/Notifications).
- **Layer 2 (Processing):** Kotlin Parsing Module + Regex.
- **Layer 3 (Storage):** Encrypted SQLite (SQLCipher).
- **Layer 4 (UI):** React Native / Expo UI.

### 4.2 Database Schema (Core Tables)

Table: <b>transactions</b>	Type	Description
id	UUID	Internal primary key
tx_id	String	Bank-issued transaction ID (unique)
amount	Decimal	Negative for Debit, Positive for Credit
bank_name	Enum	CBE, Telebirr, BOA, Awash, etc.
category	String	Food, Transport, etc.
status	Enum	Detected, Verified, Flagged

raw_metadata	JSO N	Non-sensitive parsed fragments
--------------	----------	--------------------------------

### 4.3 Data Privacy Protocol

LedgerOne must operate under a **Zero-Cloud Architecture** for the MVP.

- **Encryption:** The database key is stored in the **Android Keystore/iOS Keychain**.
  - **No PII:** The app does not ask for Account Numbers or Full Names.
- 

## 5. UI/UX Design System

### 5.1 Design Principles

1. **Density over Fluff:** Financial users want to see more data at once. Use tight lists, not large cards.
2. **Immediate Feedback:** When a transaction is detected, a "Low-Intrusion" Snackbar should appear.
3. **The "Confidence" Badge:** Use a green checkmark for Verified receipts to build trust.

### 5.2 Key Screens Detail

- **Onboarding:** A 3-step wizard explaining *why* we need SMS permissions.
  - **The Vault:** A chronological list of all transactions with a powerful search/filter (Filter by Bank, Date Range, Category).
- 

## 6. Compliance & Play Store Strategy

The **READ\_SMS** permission is highly scrutinized. To prevent rejection:

1. **Privacy Policy:** Must explicitly state that data is **only** processed locally and never sold.
  2. **Prominent Disclosure:** A full-screen modal *before* the system permission prompt explaining that "LedgerOne uses SMS to automate your expense tracking."
  3. **Video Demonstration:** A mandatory requirement for Google—we must record the app detecting a transaction in real-time.
- 

## 7. Monetization Strategy

### 7.1 Freemium Model

- **Free:** Track up to 2 banks, 1 month of history, basic categories.

- **Premium (Monthly/Annual):** Unlimited banks, PDF Export (for tax/business), custom categories, and Spending Alerts.

## 7.2 Local Payment Integration

Since this is for Ethiopia, the premium upgrade must be payable via **Telebirr** and **CBE Birr** directly within the app or via a Telegram bot.

---

## 8. Risk Management

Risk	Severity	Mitigation
<b>Bank SMS Format Change</b>	High	Remote Regex updates via JSON config.
<b>Permission Denied</b>	High	Provide a "Manual Entry" fallback that is so good users eventually want to automate it.
<b>Battery Optimization</b>	Medium	Request "Ignore Battery Optimizations" to keep the listener alive.

---

## 9. 17 Roadmap (The 12-Month Plan)

- **Month 1-2 (MVP):** Core parsing for CBE and Telebirr. Dashboard.
  - **Month 3-4 (Expansion):** BOA, Awash, Dashen support. Receipt verification engine.
  - **Month 6 (Intelligence):** Budgeting modules and AI-based spending predictions.
  - **Month 12 (B2B):** "LedgerOne for Business" — Exporting data to accounting software like QuickBooks.
- 

## 10. Success Metrics (KPIs)

- **Retention (D30):** 25% of users who download must be active after 30 days.

- **Automation Ratio:** 90% of a user's transactions should be captured without manual typing.
- **Verification Rate:** 50% of SMS transactions should be successfully "Verified" via URL scraping.
-