

Product Requirements Document:

LedgerOne (v1.0)

Status: Draft | Target Launch: Q3 2026 | Owner: Product Lead

1. Executive Summary

LedgerOne is a decentralized personal finance manager (PFM) specifically engineered for the Ethiopian ecosystem. By leveraging **on-device parsing** of SMS and push notifications, it aggregates fragmented financial data from providers like CBE, Telebirr, and Awash into a single, actionable dashboard. It solves the "Open Banking Gap" by treating the smartphone's notification tray as the data gateway.

2. User Personas & Stories

2.1 Personas

Persona	Profile	Core Need
The Freelancer	Works for international clients; uses BOA and Telebirr.	Needs to track exchange rates and total liquidity.
The Merchant	Small shop owner using Telebirr for 50+ daily txns.	Needs to verify payments quickly without scrolling SMS.
The Professional	Salary earner at a private firm.	Needs to know "Where did my money go?" by month-end.

2.2 User Stories

- **As a user,** I want the app to automatically detect a CBE Birr SMS so that I don't have to manually enter my lunch expense.
- **As a user,** I want to verify a transaction via the bank's official web link so that I can be certain the SMS wasn't spoofed.

- **As a user**, I want my financial data to stay on my phone so that my privacy is never compromised.
-

3. Functional Requirements

3.1 Ingestion & Parsing Engine

The engine must identify patterns in unstructured text.

- **Pattern Matching:** Use localized Regex for Ethiopian banks (e.g., `Amt: {amount} ETB, From: {sender}`).
- **Notification Listener Service:** Must run as a background service to intercept push notifications from banking apps (CBE Mobile, Telebirr App).
- **Duplicate Detection:** Logic to ensure that if a transaction triggers both an SMS and a Notification, it is logged as a single event based on the `TransactionID`.

3.2 Receipt Verification System

1. **Extraction:** Identify URLs within the SMS body.
2. **Whitelisting:** Only allow domains from recognized banks (e.g., [https://www.cbe.com.et/...](https://www.cbe.com.et/)).
3. **Headless Scraping:** Fetch the HTML, extract the transaction status, and match the `TransactionID`.
4. **State Update:** Change status from `Detected` to `Verified`.

3.3 The "Universal Dashboard"

- **Liquidity View:** Aggregated balance across all detected accounts.
 - **Cash Flow:** A bar chart comparing "Inflow" vs. "Outflow" for the current month.
 - **Smart Categorization:**
 - **Ruleset:** If Sender = "8099" (Telebirr) and Text contains "Airtime", Category = "Utilities".
 - **Manual Override:** Users can re-categorize; the app "learns" from this locally.
-

4. Technical Architecture & Data Security

4.1 Tech Stack

- **Frontend:** React Native (High performance for both Android/iOS).
- **Local Database:** **SQLCipher** for 256-bit AES encryption of the SQLite database at rest.
- **Logic:** Kotlin-based Android Fragments for high-level SMS/Notification permissions handling.

4.2 Security Protocol (Zero-Knowledge)

Constraint: No financial data shall be transmitted to LedgerOne servers.

- **Local Processing:** All Regex parsing happens in the app's private memory space.
 - **Anonymized Analytics:** Only usage data (e.g., "User clicked Dashboard") is sent to Telemetry; no transaction amounts or IDs.
-

5. Permission Strategy & Compliance

Google Play has strict policies regarding `READ_SMS`. To ensure approval:

1. **The "Financial SMS" Exception:** We must declare the app under the "Financial Services" category and use the `SMS_FINANCIAL_TRANSACTIONS` declaration.
 2. **Progressive Disclosure:**
 - **Initial Launch:** Ask for **Notification Access** (less scary, highly effective).
 - **Contextual Prompt:** When a user sees an unverified transaction, prompt: "*Want 100% accuracy? Enable SMS parsing for deep verification.*"
-

6. Non-Functional Requirements

- **Performance:** Parsing an incoming SMS must take \$< 200ms\$ to avoid UI lag.
 - **Reliability:** The background listener must have an "Auto-restart" capability if killed by the OS (Android Battery Optimization).
 - **Scalability:** The parsing module must be "Adapter-based" to allow adding new banks (e.g., Hijra Bank) via a remote JSON config update without a full app store release.
-

7. Monetization & Growth

7.1 Pricing Model

- **LedgerOne Basic:** Free. Includes 2 banks, basic categorization, 30-day history.
- **LedgerOne Pro:** 79 ETB/month. Includes unlimited banks, PDF export for taxes, and "Spending Predictions."

7.2 The "Trust" Hook

Since this is a high-sensitivity app, the first 1,000 users should get a "**Privacy First**" digital badge and a lifetime discount. Marketing should focus on "**Your bank data, staying on your phone.**"

8. Success Metrics (KPIs)

- **Onboarding Success:** % of users who grant at least one permission (Target: > 70%).
 - **Categorization Accuracy:** % of transactions correctly labeled without user intervention (Target: > 85%).
 - **Retention:** Day 30 Retention (Target: > 30% — high for PFM apps).
-

9. Roadmap

- **V1 (MVP):** CBE & Telebirr support, basic dashboard, local encryption.
- **V1.5:** BOA, Awash, and Dashen support; PDF export.
- **V2.0:** Budgeting tools, cross-device sync (encrypted), and AI-driven "Savings Tips."
-