

Bank-Grade Reconciliation AI - Project Documentation

1. Project Overview

The Bank-Grade Reconciliation AI is a specialized multi-agent system designed to automate financial reconciliation between invoices and bank payments. It leverages Oracle 23AI's converged database capabilities, including Vector Search and JSON storage, to handle semi-structured remittance data.

2. System Architecture

- Frontend: Streamlit dashboard for monitoring and manual intervention.
- Backend: FastAPI orchestrator managing the agent lifecycle.
- Database: Oracle 23AI as a unified relational, JSON, and Vector store.
- Integration: RESTful communication between UI and Backend services.

3. The Agentic AI Pillars

Unlike traditional automation, this solution is 'Agentic' because it exhibits:

- Specialization: Independent units (Agents) with specific domain logic (Intake, Matching, Audit).
- Reasoning: The ability to interpret messy 'dirty' remittance text and find semantic similarities.
- State Management: An orchestrator that manages a complex lifecycle and handles handovers.
- Goal-Orientation: Moving toward a 'High Confidence Match' rather than just executing rigid scripts.

4. Primary Use Cases

A. Ambiguous N:1 Bulk Matching:

Resolving cases where a single large payment references multiple partial invoice codes buried in unstructured text. The AI extracts references, sums invoice totals, and validates the match.

Bank-Grade Reconciliation AI - Project Documentation

B. Intelligent Exception Classification:

Identifying 'Short Payments' (Deductions) hidden in text, such as early payment discounts or disputed items, and classifying them correctly for manual review.

5. Human-in-the-Loop & Governance

The system explicitly supports a 'Human-in-the-Loop' model. When confidence scores drop below pre-defined thresholds, the AI flags the transaction as an 'Outlier.' Finance operators can then:

- Manually link invoices to payments via the UI.
- Provide a structured justification (Reasoning).
- Maintain a bank-grade Audit Trail where every human decision is logged, versioned, and immutable.

6. Technical Stack Summary

- Language: Python (FastAPI, Streamlit, Pydantic)
- Database: Oracle 23AI (SQL, VECTOR, JSON)
- AI Models: OpenAI/Gemini for semantic extraction and reasoning
- Security: Environment-based credential management (.env) and multi-tenant isolation.