🏦 Bank Reconciliation Agent – Executive One-Pager

Last Updated: 2025-07-25 04:04:29

# 📌 Purpose

To automate the preparation and reconciliation of bank transactions into Xero from multiple file types (CSV, Excel, PDF, OFX), including the application of rules-based logic and future ML capabilities for smarter account allocation and matching.

# 📂 Key Features

- Upload interface via FastAPI Swagger UI

- Bank-specific parsers (NBB, KFH Account, KFH Card)

- Smart header detection and sign logic

- BHD conversion and cleansing of foreign currency suffixes

- Preparation of Xero bank import templates

# 📈 Architecture & Tech Stack

- FastAPI + Uvicorn backend

- File routing via filename logic

- Parser engine using Pandas

- Google Sheets integration and future Xero API (planned)

# 🧠 Next Phase

• Map to Xero template and enable download

• Remittance parsing and invoice allocation

• ML prediction model for account coding

• Full reconciliation recommendation engine

Bank Reconciliation Agent – Project Summary

Last Updated: 2025-07-25 03:54:49

# 🔍 Overview

The Bank Reconciliation Agent automates the preparation and reconciliation of bank transactions into Xero. It parses various statement formats (CSV, Excel, PDF, OFX), normalizes and transforms the data into a Xero-importable format, and is being developed to support machine learning recommendations and invoice allocation using remittance files.

# ✅ Completed

- FastAPI server and route structure setup.

- Live Swagger UI for file upload testing.

- Upload endpoint `/upload/bank-statement` implemented.

- Parsers created for:

* • NBB CSV
* • KFH Account Excel
* • KFH Card Excel

- Smart header row detection using keyword matching.

- Logic to detect sign using Debit/Credit fields.

- Use of 'BHD Equivalent' for foreign currency handling.

- bank\_accounts.json tracks accounts to process (e.g., ignores auto-fed ZAR account).

# 📁 Project Structure

Base Directory:

C:\Users\Finance Manager\OneDrive\The Botes Family\Johann's Documents\Agents\Bank Recon Agent

Subfolders and Files:

* • bank\_reco\_agent\app\main.py – App entrypoint
* • bank\_reco\_agent\app\routes\upload.py – Upload route handler
* • bank\_reco\_agent\app\services\parser.py – All parsing logic
* • bank\_reco\_agent\app\config\bank\_accounts.json – Tracked accounts to exclude/include

# 🧠 Core Logic

• Parses Excel, CSV, PDF (preview only), OFX (skipped).

• Cleans up currency strings, removes suffixes (e.g., 'BHD').

• Identifies amounts from 'BHD Equivalent' where applicable.

• Sign inferred by which column (debit or credit) is filled.

• Bank-specific parsing functions automatically selected by filename.

# 📂 Test Files

- NBB Transactions 15May25.csv

- KFH Account Statement 15May25.xlsx

- KFH Card Statement 15May25.xlsx

- OFX file (Investec – excluded from parsing)

- PDF file (for preview only)

# 🚀 Next Steps

1. Map parsed transactions to Xero CSV template for download.

2. Add `/upload/remittance` route for invoice breakdowns.

3. Apply invoice allocation logic based on remittance Excel files.

4. Build reconciliation logic using ML or rules-based mapping.

5. Integrate Xero API to auto-post transactions (optional future phase).

**Recommended Architecture**

You're already using a solid stack — here's what I'd recommend, with notes on where what you have fits and where something else might help.

| **Component** | **Recommendation** | **Notes** |
| --- | --- | --- |
| **Backend Framework** | ✅ **FastAPI** | Stick with this. Perfect for handling file uploads, API endpoints, and async processing. |
| **Agent Logic** | ✅ **Python + LangChain or OpenAI tools** | Use Python to orchestrate, and LangChain if you want reusable chains or agents for extraction, matching, or learning. |
| **PDF/Excel Parsing** | ➕ **pdfplumber / tabula-py (PDF)** **pandas / openpyxl / xlrd (Excel)** | These are gold for structured extraction. OCR (e.g. Tesseract) only if PDF is scanned. |
| **Machine Learning** | ➕ **Simple supervised ML using scikit-learn or LightGBM** | You can train a model on past reconciled transactions (description → account). Use vector embedding if going smart (e.g. OpenAI Embeddings + similarity matching). |
| **Storage** | ✅ **Google Sheets + Xero API**  ➕ Optionally: **PostgreSQL or Supabase** | Sheets + Xero will be enough for now. Add DB if you need historical learning, versioning, or caching later. |
| **UI / Chatbot** | ✅ **Google Chat + Flowise** (for simple workflows)  OR ➕ **Streamlit or Gradio** (for interactive recon dashboards) | Google Chat for quick Q&A, but add a proper web UI if you want drag-drop, preview, or override controls. |
| **File Handling** | ✅ Current setup with attachments | Works fine, but ensure max file size handling and PDF/image OCR fallback. |
| **Deployment** | ✅ **Docker + Render** | Keep going with this stack. Add Celery + Redis for async tasks later if needed. |
| **Learning from Feedback** | ➕ Store feedback (manual corrections) to improve future predictions | Could be as simple as logging changes in a sheet/CSV or storing in DB to retrain monthly. |

**Bank Reconciliation Agent – Visual Flowchart**

pgsql

CopyEdit

┌─────────────────────────────┐

│ File Upload (PDF/Excel) │ ◄─── User uploads bank statements

└────────────┬────────────────┘

│

▼

┌─────────────────────────────┐

│ File Type Detection │

│ + Extract Transactions │

│ (pdfplumber, pandas, etc.)│

└────────────┬────────────────┘

│

▼

┌─────────────────────────────┐

│ Standardize to Xero Format │

│ (account, date, desc, amt) │

└────────────┬────────────────┘

│

▼

┌─────────────────────────────┐

│ ML Classifier Lookup │◄──── Historic matched data

│ (Text → Account Suggestion)│

└──────┬─────────┬────────────┘

│ │

│ ▼

│ Ask User if Unclear

│ (Google Chat Bot)

▼

┌─────────────────────────────┐

│ Finalized Transactions File │

│ (CSV ready for Xero import) │

└────────────┬────────────────┘

│

▼

┌─────────────────────────────┐

│ Reconciliation Suggestion │

│ (Match to Contacts/Invoices)│

└────────────┬────────────────┘

▼

┌─────────────────────────────┐

│ Learn from Corrections/User │

│ + Save Feedback to Model DB │

└─────────────────────────────┘

**Folder/Code Structure (FastAPI + ML + Google Chat)**

bash

CopyEdit

bank\_reco\_agent/

├── app/

│ ├── main.py # FastAPI entry point

│ ├── routes/

│ │ ├── upload.py # Handles file upload endpoints

│ │ └── reconcile.py # Handles reconciliation & ML prediction

│ ├── services/

│ │ ├── parser.py # Extract from PDF/Excel to DataFrame

│ │ ├── formatter.py # Convert to Xero import template

│ │ ├── classifier.py # ML model inference

│ │ └── feedback.py # Save user corrections for learning

│ ├── utils/

│ │ ├── file\_utils.py # Helpers for handling file types

│ │ ├── ocr.py # Optional OCR (Tesseract wrapper)

│ │ └── xero\_format.py # Xero-specific formatting rules

│ └── models/

│ └── classifier.joblib # Trained classifier model

├── data/

│ ├── training\_data.csv # Transactions with correct accounts

│ └── feedback\_log.csv # New corrections for retraining

├── tests/

│ └── test\_parser.py # Unit tests

├── docker/

│ └── Dockerfile # Deployment container

├── requirements.txt

└── README.md