Transaction Reconciliation Tool - Requirements Document

Project: Transaction Reconciliation Tool
Date: June 27, 2025
Version: 1.0

Executive Summary

The Transaction Reconciliation Tool is a comprehensive web application designed to automate the comparison and analysis of transaction data between internal systems and external provider statements. The application leverages artificial intelligence for intelligent column mapping and anomaly detection, providing users with an intuitive interface for uploading, processing, and analyzing CSV transaction files.

Functional Requirements

File Upload and Validation

- **Drag-and-Drop Interface**: Support for intuitive file upload with visual feedback
- **File Format Support**: CSV files only, with automatic format validation
- **Size Limitations**: Maximum 10MB per file with real-time size checking
- **Progress Indicators**: Visual feedback during upload and processing
- **Error Handling**: Clear error messages for invalid files or upload failures

Transaction Comparison Engine

- **Automated Matching**: Compare transactions based on unique transaction_reference field
- **Categorization**: Sort results into three categories:

- Matched Transactions (present in both files)
- Internal Only (present only in internal system)
- Provider Only (present only in provider statement)
- **Discrepancy Detection**: Identify amount and status mismatches in matched transactions
- **Data Integrity**: Ensure accurate comparison with proper data type handling

User Interface and Dashboard

- **Responsive Design**: Mobile-friendly interface compatible with desktop and mobile devices
- **Summary Statistics**: Display transaction counts and percentages for each category
- **Data Visualization**: Interactive pie charts and bar graphs for data distribution
- **Advanced Tables**: Sortable, filterable, and paginated transaction views
- **Search Functionality**: Real-time search across transaction data
- **Visual Highlighting**: Color-coded rows based on transaction type and discrepancies

AI-Powered Enhancements

- **Intelligent Column Mapping**: Automatic detection and mapping of CSV column headers
- Support for common field variations (id, reference, txn_id, transaction_id)
- Confidence scoring for mapping accuracy
- Fuzzy matching for similar column names
- **Anomaly Detection**: Machine learning-based identification of suspicious transactions
- Isolation Forest algorithm for outlier detection
- Statistical variance analysis for amount discrepancies
- Pattern recognition for status conflicts

- **Risk Assessment**: Automatic categorization of transactions by risk level (Low, Medium, High)

Export and Reporting

- **Multiple Export Options**: Client-side and server-side CSV generation
- **Category-Specific Exports**: Individual CSV files for each transaction category
- **Bulk Export**: ZIP file containing all reconciliation results
- **Data Preservation**: Maintain all original data plus calculated fields in exports

Technical Requirements

Frontend Specifications

- **Framework**: React 18 with modern hooks and functional components
- **Styling**: TailwindCSS utility-first framework with shadcn/ui components
- **Data Visualization**: Recharts library for interactive charts and graphs
- **File Handling**: React Dropzone for upload interface, PapaParse for CSV processing
- **HTTP Communication**: Axios for API requests with proper error handling
- **Browser Support**: Modern browsers with ES6+ support

Backend Specifications

- **Framework**: Flask web framework with RESTful API design
- **Data Processing**: Pandas library for efficient CSV manipulation and analysis
- **Machine Learning**: Scikit-learn for anomaly detection algorithms
- **Cross-Origin Support**: Flask-CORS for frontend-backend communication
- **File Security**: Werkzeug for secure file upload handling
- **Session Management**: In-memory caching for demo purposes

Performance Requirements

- **File Processing**: Handle CSV files up to 10MB with optimal performance under 5MB
- **Response Time**: Complete reconciliation within 30 seconds for typical datasets
- **UI Responsiveness**: Interactive elements respond within 100ms
- **Export Speed**: Generate and deliver CSV exports within 10 seconds
- **Memory Efficiency**: Process large datasets without memory overflow

System Assumptions

Data Format Assumptions

- CSV files contain unique transaction_reference values within each file
- Files are well-formed with proper headers and consistent data types
- Standard fields include transaction_reference, amount, and status at minimum
- All monetary amounts use the same currency for comparison purposes
- Date formats are consistent within each file

Operational Assumptions

- Single-user demonstration environment without concurrent access
- Uploaded files overwrite previous uploads (no persistent storage)
- No user authentication or authorization required
- Session-based data isolation sufficient for demo purposes
- Network connectivity available for frontend-backend communication

Technical Assumptions

- Modern web browser with JavaScript enabled
- Sufficient client-side memory for file processing
- Stable network connection for file uploads

- Python 3.11+ and Node.js 20+ available for deployment

Project Scope

Included Features

- Web-based user interface for file upload and result visualization
- Backend API for CSV processing and transaction reconciliation
- AI-powered column mapping with confidence scoring
- Machine learning-based anomaly detection and risk assessment
- Export functionality for reconciliation results in multiple formats
- Responsive design optimized for desktop and mobile devices
- Real-time validation and comprehensive error handling
- Interactive data tables with advanced sorting and filtering
- Summary statistics and data visualization charts
- Visual indicators for different transaction types and anomalies

Excluded Features

- User authentication and authorization system
- Persistent data storage or database integration
- Multi-user support or advanced session management
- Complex reporting or analytics beyond basic reconciliation
- Integration with external payment or banking systems
- Automated scheduling or batch processing capabilities
- Advanced security features beyond basic file validation
- Support for file formats other than CSV
- Real-time data synchronization or live updates
- Advanced audit trails or compliance reporting

AI Feature Specifications

Column Mapping Intelligence

The system employs natural language processing techniques to automatically identify and map CSV column headers to standardized field names. The mapping algorithm uses keyword matching with confidence scoring to handle variations in column naming conventions.

Supported Field Mappings: - Transaction Reference: id, reference, txn_id, transaction_id, ref, txn_ref, trans_id - Amount: amount, value, total, sum, price, cost, fee, charge - Status: status, state, condition, stage, phase - Date: date, time, timestamp, created, processed - Currency: currency, curr, ccy

Anomaly Detection System

The application implements multiple machine learning algorithms to identify suspicious or unusual transactions that require manual review.

Detection Methods: - **Isolation Forest**: Unsupervised learning algorithm for outlier detection in transaction amounts - **Statistical Analysis**: Variance calculation for amount discrepancies exceeding 5% threshold - **Pattern Recognition**: Rule-based detection of critical status mismatches (e.g., Processed vs Failed)

Risk Classification: - **Low Risk**: Standard transactions with minor or no discrepancies - **Medium Risk**: Transactions with moderate anomalies detected by ML algorithms - **High Risk**: Transactions with significant amount variances or critical status conflicts

User Experience Design

Workflow Process

1. User accesses the web application through a modern browser
2. Drag-and-drop or click to upload two CSV files (internal and provider)
3. System validates files and displays upload progress
4. AI performs automatic column mapping with results shown to user

5. Backend processes files and performs transaction reconciliation
6. Results displayed in categorized tables with visual highlighting
7. AI insights presented in dashboard with anomaly detection results
8. User reviews results and exports data in preferred format

Visual Design Principles

- **Clarity**: Clean, uncluttered interface with clear visual hierarchy
- **Consistency**: Uniform design patterns and color schemes throughout
- **Accessibility**: High contrast colors and keyboard navigation support
- **Responsiveness**: Adaptive layout for various screen sizes and devices
- **Feedback**: Immediate visual feedback for user actions and system status

Color-Coded Indicators

- **Green**: Perfect matches with no discrepancies
- **Orange**: Amount mismatches requiring attention
- **Red**: Status conflicts or provider-only transactions
- **Purple**: AI-detected anomalies flagged for review
- **Yellow**: Internal-only transactions not found in provider data

Quality Assurance

Testing Requirements

- **Unit Testing**: Individual component and function testing
- **Integration Testing**: API endpoint and data flow validation
- **User Interface Testing**: Cross-browser compatibility and responsive design
- **Performance Testing**: Load testing with various file sizes
- **Security Testing**: File upload validation and input sanitization

Acceptance Criteria

- All functional requirements implemented and tested
- AI features demonstrate accurate column mapping and anomaly detection
- Export functionality produces correct and complete data files
- User interface provides intuitive and responsive experience
- Application handles edge cases and errors gracefully
- Performance meets specified targets for file processing and response times

Deployment and Maintenance

Deployment Options

- **Development**: Local environment with Flask development server and Vite
- **Production**: Multiple deployment strategies supported
- Replit for integrated full-stack deployment
- Vercel (frontend) + Fly.io (backend) for scalable architecture
- Single Flask deployment with integrated React build

Maintenance Considerations

- Regular updates to machine learning models for improved accuracy
- Monitoring of file processing performance and optimization
- Security updates for dependencies and frameworks
- User feedback collection and feature enhancement planning

This document serves as the comprehensive requirements specification for the Transaction Reconciliation Tool, ensuring all stakeholders have a clear understanding of the project scope, technical specifications, and expected deliverables.