Software Requirements Specification (SRS)

Fintech Document Parsing & Expense Intelligence Platform

(IEEE-830 / IEEE-9000 Style)

DevGeeks

November 29, 2025

Contents

1 Introduction

1.1 Purpose

This document describes the **functional and non-functional requirements** for the Fintech Document Parsing & Expense Intelligence Platform (hereafter referred to as "the System").

The System allows users to:

- Upload bank statements (PDFs), photos of bills, and other financial documents.
- Automatically parse, extract, categorize, and store financial data.
- Generate CSV/Excel outputs of transactions.
- Produce **AI-generated financial insights** and trend analytics.
- Query their own documents using **RAG-based conversational search**.
- View structured reports, charts, and spending trends.

The SRS is intended for:

- Developers
- Architects
- QA engineers
- Stakeholders
- Compliance/security reviewers

1.2 Scope

The System is a multi-module fintech solution that:

- Ingests financial documents (PDF, images).
- Uses a multi-stage parsing pipeline (Regex → BERT → optional LLM fallback).
- Stores structured data in per-user buckets and relational DB.
- Supports RAG for user-specific document searches.
- Provides analytics dashboards and downloadable reports.
- Enables AI-based insights (monthly summaries, spending breakdowns).

The System will have both:

- **End-user plan** (individual users)
- Business plan (advanced accounting & forecasting) — *future scope.*

This SRS covers **end-user plan functionality only**.

1.3 Definitions, Acronyms, Abbreviations

Term	Definition
PDF Parser	Component that extracts text/tables from PDFs.
OCR	Optical Character Recognition for converting images to text.
RAG	Retrieval-Augmented Generation using vector search + LLM.
BERT	Lightweight local ML classifier used for expense categorization.
LLM	Large Language Model (Gemini/ChatGPT/R1) used as fallback.
Transaction	Parsed financial entry: amount, date, vendor, category, etc.
Bucket	Per-user storage namespace for documents and outputs.
Structured Data	Parsed tables stored in relational DB.

1.4 References

1. IEEE Computer Society, "IEEE Std 830-1998 – Software Requirements Specification."
2. PCI-DSS v4.0 standards.
3. GDPR/CCPA privacy guidelines.
4. Internal architectural diagrams (DFD1, DFD2).

2 Overall Description

2.1 Product Perspective

The system is a **standalone SaaS platform** with:

- A web/mobile frontend.
- Backend microservices for parsing, classification, analytics.
- Object storage for user files.
- Relational database for structured financial data.
- Vector store for RAG queries.

It interfaces with:

- 3rd-party OCR engines (optional).
- 3rd-party LLM APIs (optional fallback).
- Internal ML models (BERT classifier).

2.2 Product Functions

Major system functions:

- **Document Upload**
  - Accept PDFs, images, Excel files, bills, receipts.
- **Data Extraction Pipeline**
  - PDF parsing.
  - OCR for images.
  - Regex-based initial classification.
  - BERT classification.
  - LLM fallback classification when needed.
- **Data Storage**
  - Raw documents in per-user buckets.
  - Parsed transactions in relational tables.
  - Embedded vectors for RAG.
- **Analytics & AI Insights**
  - Monthly/annual spending summary.
  - Category-wise charts.
  - Vendor analysis.
  - Custom AI insights using aggregated data.
- **RAG-based conversational interface**
  - Query spending.
  - Query document contents (policies, bills, etc.).
- **Report Generation**
  - CSV, Excel, JSON summaries.
  - Visual charts.

2.3 User Classes and Characteristics

User Class	Description	Technical Level
End Users	Individuals uploading their financial documents	Low
Admin	Supports debugging, user management, audit logs	Moderate
Business Users (future)	Accounting / finance professionals	High

2.4 Operating Environment

- **Backend:** Cloud-native (AWS/GCP/Azure)
- **Frontend:** Web/mobile browser
- **API:** HTTPS REST + WebSocket optional
- **Storage:** S3/GCS-style bucket, SQL database
- **Vector Search:** Pinecone/PGVector/Weaviate/etc.

2.5 Design and Implementation Constraints

- Bank-grade security (TLS1.2+).
- PCI-DSS compliant storage if card data appears (rare).
- Vendor lock-in minimization (abstract OCR & LLM providers).
- Maximize local computation to minimize LLM cost.

2.6 User Documentation

- Web-based documentation.
- Onboarding wizard.
- API documentation for business plan.

2.7 Assumptions & Dependencies

Assumptions:

- PDFs are legible and follow common banking statement formats.
- OCR quality is adequate for bill photos.

Dependencies:

- OCR provider uptime.
- Cloud storage.
- LLM provider for fallback classification/insights.

3 System Features

3.1 Feature: User Document Upload

3.1.1 Description

Users upload PDFs, images, and financial documents to the System.

3.1.2 Functional Requirements

ID	Requirement
FR-1	The system shall allow users to upload PDFs, JPEG/PNG images, and Excel files.
FR-2	The system shall validate file type and size.
FR-3	The system shall store raw files in the user's bucket.
FR-4	The system shall generate a document ID for tracking.
FR-5	The system shall extract metadata such as upload time and user ID.

3.2 Feature: Document Parsing Pipeline

3.2.1 Description

Extracts text, tables, and transactions using a multi-stage fallback mechanism.

3.2.2 Functional Requirements

ID	Requirement
FR-10	The system shall parse PDFs using layout-aware extraction.
FR-11	The system shall perform OCR for images.
FR-12	The system shall map parsed content into a canonical transaction schema.
FR-13	The system shall categorize transactions using Regex rules.
FR-14	The system shall pass non-classified transactions to a local BERT model.
FR-15	The system shall use an LLM fallback when BERT confidence is below a defined threshold.
FR-16	The system shall store the classification source (Regex, BERT, LLM) and confidence score.

3.3 Feature: Transaction Storage & Management

ID	Requirement
FR-20	The system shall store all parsed transactions in a relational DB.
FR-21	The system shall index transactions by user, date, vendor, and category.
FR-22	The system shall allow regeneration of CSV/Excel reports from the DB.
FR-23	The system shall flag low-confidence classifications for user review.

3.4 Feature: AI Insights & Reporting

ID	Requirement
FR-30	The system shall generate monthly spending summaries.
FR-31	The system shall compute category totals, trends, and vendor statistics.
FR-32	The system shall generate AI-written financial insights based on aggregated data.
FR-33	The system shall store generated reports as JSON/HTML in user buckets.
FR-34	The system shall allow export of summaries and statistics to CSV/Excel.

3.5 Feature: RAG-based Conversational Queries

3.5.1 Functional Requirements

ID	Requirement
FR-40	The system shall embed unstructured documents (text chunks) into a vector store.
FR-41	The system shall isolate embeddings per user for security.
FR-42	The system shall interpret natural language queries using an LLM.
FR-43	The system shall perform SQL queries for structured-data questions (e.g., "How much did I spend on groceries in Q3?").

FR-44	The system shall use RAG retrieval for unstructured questions (e.g., "What is the policy for expense reports in this document?").
FR-45	The system shall produce natural language answers with citations to the source documents.

3.6 Feature: Charts & Visualization

ID	Requirement
FR-50	The system shall provide category-wise spending graphs.
FR-51	The system shall generate vendor-level spending charts.
FR-52	The system shall display monthly/quarterly/yearly trends.
FR-53	The system shall support downloadable PNG/SVG graphs.

4 External Interface Requirements

4.1 User Interface

The UI shall include, but not be limited to, the following screens/views:

- File upload screen
- Document parsing progress view
- Insights dashboard
- Charts & analytics view
- Chat interface (ask-your-data)
- Report export/download page

4.2 APIs

The system shall expose the following primary REST API endpoints:

- `POST /upload`
- `GET /transactions`
- `GET /reports/monthly`
- `POST /chat/query`
- `GET /charts/category`

4.3 Hardware Interfaces

N/A (The platform is cloud-native and does not require specific end-user hardware interfaces beyond standard web/mobile devices).

4.4 Software Interfaces

- OCR provider API (e.g., Amazon Textract, Google Vision)
- Cloud storage API (e.g., S3, GCS)
- LLM provider API (e.g., Gemini, ChatGPT)
- Vector DB API (e.g., Pinecone, PGVector)

4.5 Communication Interfaces

- All communication uses **HTTPS**.
- **JWT** authentication shall be used for API access.

5 Non-Functional Requirements

5.1 Performance Requirements

ID	Requirement
NFR-1	PDF parsing shall complete within **10 seconds** for a 20-page statement.
NFR-2	OCR results shall return within **5 seconds** for a typical bill image.
NFR-3	Transaction classification pipeline shall process at least **1000 txns/min**.
NFR-4	Chat queries shall respond within **1–3 seconds** (cached) or **5–8 seconds** (RAG+LLM).

5.2 Security Requirements

ID	Requirement
NFR-10	All data in transit shall use **TLS 1.2+**.
NFR-11	All stored data shall be **encrypted at rest**.
NFR-12	Embeddings and financial data shall be **isolated per user**.
NFR-13	Admins shall not access user financial content without explicit, auditable permission.
NFR-14	Audit logs must track key actions (upload, parse, modify, admin access).

5.3 Reliability & Availability

- The System shall target a **99.5% uptime** availability.
- Unexpected parsing errors must fall back to safer modules or flag for manual review.
- A robust retry mechanism for external dependencies (OCR, LLM calls) must be implemented.

5.4 Maintainability

- The system must implement a **modular parsing pipeline** (clear separation between OCR, extraction, classification).
- Configurable regex rules must be stored in DB/table for easy update.
- Tools for rapid embedding regeneration must be available.

5.5 Portability

- The architecture shall be designed to be **cloud-agnostic**.
- All primary models and microservices must be **containerized (Docker)**.

6 Other Requirements

6.1 Legal & Compliance

- Must comply with GDPR/CCPA user deletion rights.
- Financial data must be classified as sensitive data and handled accordingly.
- No sharing of user data with third-party LLMs without explicit user permission.

6.2 Future Enhancements (Business Plan)

The following features are not part of the current release scope but are planned for future versions:

- Accounting automation (e.g., auto-reconciliation).
- PowerBI/Tableau exports.
- Asset valuation dashboard.
- Project estimation (NPV, IRR, DCF).
- Multi-user/shared workspace functionality.

7 Appendices

7.1 Appendix A – DFDs

- DFD1 - High-Level System Context Diagram
- DFD2 - Document Parsing Detailed Flow

(Insert DFD visual representations here.)

7.2 Appendix B – Sample Document Schemas

- Transaction schema (fields: transaction_id, user_id, date, amount, vendor, category, confidence, source_doc_id)
- Report schema
- Embedding metadata schema