Software Requirements Specification (SRS)

Project: **Lab Activity 14**

Team Name: **Tommy W., Nadia H., Damarion M.P, Samantha V.M, Stephany C.P**

Group Number: **4**

Version: **4.0**

Date: **December 2025**

Contents

Version Description

Version	Description	Date
1.0	Initial SRS including introduction, interface requirements, functional and non-functional requirements, and glossary.	December 5, 2025
2.0	Added functional and non-functional requirement updates for Snapshot 2.	December 7, 2025
3.0	Expanded system workflows and clarified interface behaviors for Snapshot 3.	December 9, 2025
4.0	Final refinements, legal updates, and completed requirement definitions for Snapshot 4.	December 10, 2025

1 Introduction

1.1 Purpose of the Document

The purpose of this Software Requirements Specification (SRS) is to define the functional and non-functional requirements for the Intelligent Document Processing & Case File Search System. The software enables users to upload PDF case files, extract simulated metadata, store it in a PostgreSQL database, and search stored records. This document serves as a guideline for developers, instructors, and future maintainers.

1.2 Intended Audience

This SRS is intended for:

- Developers implementing frontend, backend, and database components.
- Instructors evaluating the project's correctness and completeness.
- Students reviewing expected system behavior and features.

1.3 Overview of the Software

The system provides:

- A PDF upload interface.
- Automatic extraction of basic (simulated) metadata.
- Structured storage in a PostgreSQL database.
- A search interface to retrieve case information.

The entire system runs inside Docker containers for consistency and ease of deployment.

2 External Interface Requirements

2.1 User Interface

The software includes the following UI components:

- **Upload Page:** Allows users to select and upload PDF files.
- **Search Page:** Enables case searching by ID, name, or keywords.
- **Case Detail Page:** Displays extracted metadata for a selected case.
- **Navigation Bar:** Provides access to Home, Upload, and Search.

2.2 Software Interfaces

- **Frontend → Backend:** Communicates via REST API using JSON.
- **Backend → Database:** Interacts with PostgreSQL using environment variables.
- **Backend → Filesystem:** Temporarily stores uploaded files during processing.
- **Docker Network:** All services communicate through Docker's internal network.

2.3 Hardware Interfaces

No hardware interfaces are required. All components operate inside virtualized Docker containers.

2.4 Communication Interfaces

- HTTP-based communication through ports exposed by Docker.
- Localhost access for frontend and backend services.

3 System Requirements

3.1 Functional Requirements

FR1: Document Upload

- Users shall upload PDF files through the Upload page.
- The system shall validate file type before processing.

FR2: Metadata Extraction

- The system shall extract simulated metadata fields (case ID, parties, dates).
- The system shall return clear error messages if extraction fails.

FR3: Database Storage

- Extracted metadata shall be stored in PostgreSQL with unique identifiers.
- Duplicate file uploads shall be flagged or prevented.

FR4: Search Functionality

- Users shall search records by ID, name, or keyword.
- Matching results shall be displayed in a tabular format.

FR5: Case Detail View

- The system shall display complete metadata for any selected record.

3.2 Non-Functional Requirements

NFR1: Performance

- Metadata extraction shall complete within 5 seconds per file.
- Search queries shall return results within 2 seconds.

NFR2: Usability

- The system shall be easy to use for non-technical users.
- Navigation shall be simple and intuitive.

NFR3: Reliability

- The system shall recover automatically if any Docker container restarts.
- All containers must start reliably with `docker-compose up`.

NFR4: Security

- Uploaded documents must not be accessible outside the system.
- No data shall be sent to third-party APIs.

NFR5: Compatibility

- The system shall run on Windows, macOS, and Linux using Docker Desktop.
- The frontend shall support Chrome, Safari, and Microsoft Edge.

4 Legal and Ethical Considerations

4.1 Data Storage and Privacy

- Metadata may include sensitive information requiring careful handling.
- All data is stored locally within Docker containers.
- Documents are never sent to external services.

4.2 Ethical Use of Document Processing

- Metadata extraction must not misrepresent document contents.
- Users shall be informed when extraction is incomplete or inaccurate.

4.3 Compliance Considerations

- Only simulated test data is used—no real case files.
- The system shall follow relevant academic data policies.

5 Glossary

Acronym	Definition
UI	User Interface
API	Application Programming Interface
DB	Database
SRS	Software Requirements Specification
SDD	Software Design Document