Documentation of Skill Development Report

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Department of Computer Science and Engineering

Skill Development Lab - CO34451

Shri Govindram Seksaria Institute of Technology and Science, Indore

Submitted To:

Mr. Surendra Gupta

Ms. Mamta Gupta

Ms. Ashvini Pahade

Submitted To:

Shreyash Tiwari

0801CS233D09

Software Requirements Specifications (SRS)

For

Documatch

Version 1.0 approved

Preprated by :

Shreyash Tiwari

Organization :

Shri G.S Institute of Technology and Science

Date Created :

9.10.2024

Table of Contents

1. Introduction

1.1 Purpose

The purpose of **Documatch** is to develop a web-based application that allows users to compare documents in multiple formats (PDF, Excel, Word) and identify mismatches or discrepancies in the content, such as text, tables, or semantic differences. The system uses AI for enhanced semantic analysis and provides users with a comprehensive report on mismatches.

1.2 Document Conventions

This document follows IEEE formatting standards. Python code snippets and Flask framework conventions are included for better understanding.

1.3 Intended Audience and Reading Suggestions

The intended audience for this document includes developers, testers, project managers, and stakeholders. It provides detailed descriptions of system features, functional and non-functional requirements, and external interfaces.

1.4 Product Scope

This application automates the process of comparing documents in different formats, extracting tables and text, and identifying mismatches using both simple comparison techniques and AI-based semantic analysis. It is especially useful for organizations or individuals that need to verify data consistency across different document formats (e.g., comparing Excel tables with Word documents).

1.5 References

- Python 3.x Documentation
- Flask Framework Documentation
- Pandas Documentation
- Hugging Face Transformers Documentation
- RapidFuzz Documentation

2. Overall Description

2.1 Product Perspective

The product is a web-based system designed to automate document comparison and mismatch identification across different formats such as PDF, Excel, and Word. This application replaces the manual process of comparing documents by allowing users to upload files and receive an automatically generated report on mismatches.

2.2 Product Functions

The system provides the following functionalities:

- Upload PDF, Excel, and Word files.

- Extract tabular and text data from the documents.
- Perform AI-powered semantic comparisons and fuzzy matching for identifying mismatches.
- Present mismatches in a clear, tabular format for easy review by the user.

2.3 User Classes and Characteristics

- **General Users**: Can upload files for comparison and view mismatch reports.
- **Administrators**: Can manage system configurations (AI usage, performance tuning, etc.).

2.4 Operating Environment

The system is designed to operate on any modern web browser (e.g., Chrome, Firefox) and requires Python 3.x along with Flask for backend processing.

2.5 Design and Implementation Constraints

The system depends on several external libraries for data extraction and AI-based analysis:

- PDF extraction: PyPDF2
- Excel manipulation: Pandas
- Table comparison: Custom functions
- PDF generation: FPDF2.6 User Documentation

User manuals and tutorials will be provided as HTML and PDF files. These will explain how to upload documents, view mismatch reports, and interpret results.

2.7 Assumptions and Dependencies

- The system assumes users have access to structured data in PDF, Excel, or Word format.
- The system requires modern web browsers and Python 3.x.

3. External Interface Requirements

3.1 User Interfaces

The application will feature:

- A file upload form for selecting files to compare.
- A results page displaying mismatches in a table format with details such as row number, column name, and file comparison.
- Toggle options to enable or disable AI-based comparison.

3.2 Hardware Interfaces

No specific hardware interfaces are required; the system will run on standard desktop or mobile devices with internet access.

3.3 Software Interfaces

- **Python 3.x:** Used for the backend.
- **Flask:** Used for handling the web interface and user requests.
- **Pandas:** Used for Excel file manipulation.
- **PyPDF2:** Used for extracting text from PDFs.
- **FPDF:** Used for generating PDF files from Excel data.

3.4 Communications Interfaces

The system uses HTTP/HTTPS for communication between the web interface and backend.

4. System Features

4.1 Document Comparison

4.1.1 Description and Priority

This is the core feature, allowing users to upload documents, extract data, and compare for mismatches. Priority: High.

4.1.2 Stimulus/Response Sequences

- **User Action**: Uploads two documents for comparison.
- **System Response**: Extracts the content, performs a comparison, and displays the mismatches.

4.1.3 Functional Requirements

- **REQ-1**: The system must support file uploads for PDF, Excel, and Word files.
- **REQ-2**: The system must extract tables or text from these documents.
- **REQ-3**: The system must compare data using simple text matching and AI-powered semantic analysis.
- **REQ-4**: The system must present the comparison results in a tabular format, listing row numbers, column names, and discrepancies.
- **REQ-5**: The system must provide an option to enable or disable AI-based comparison.

5. Nonfunctional Requirements

5.1 Performance Requirements

- The system should process and compare files within 5 seconds for documents containing up to 1000 rows or 20 pages.

5.2 Security Requirements

- Secure file uploads with HTTPS must be implemented.
- User data must be protected and not shared with other users.

5.3 Usability Requirements

- The interface should be user-friendly and intuitive for non-technical users.
- The comparison results should be easy to interpret and navigate.

5.4 Maintainability Requirements

- The code should follow Python’s PEP8 guidelines for maintainability and future enhancements.

6. Other Requirements

- The system should be easily extendable to support additional document formats in the future, such as CSV or XML.

7. Glossary

- **AI-powered semantic comparison**: Using natural language processing (NLP) models to compare the meanings of text segments.
- **Fuzzy matching**: A technique used to identify strings that are approximately similar but not exactly the same.