



Low Level Design Document

Introduction

This Low Level Design (LLD) document outlines the implementation details for **DocuCluster - Hierarchical Document Clustering Tool**. DocuCluster is a web-based application for uploading text documents, performing hierarchical clustering using selectable linkage methods, visualizing dendrograms, and exporting cluster assignments. The tool is built with Flask, Scikit-learn, Pandas, and Numpy.

1. System Components

Component	Description	Key Responsibilities
Web UI	Frontend interface (HTML/JS)	Upload docs, select options, display results
Flask Backend	REST API server	Handle requests, manage workflow
Clustering Engine	Core clustering logic (Python, Scikit-learn)	Vectorize, cluster, generate dendrogram
Data Manager	Handles file I/O, data parsing, and export	Read, store, export documents/clusters

2. Class/Interface Overview

Class/Module	Key Methods/Attributes	Relationships
DocumentManager	load_files() , get_documents() , export_clusters()	Used by Flask routes
ClusteringEngine	vectorize() , cluster() , get_dendrogram()	Used by Flask routes
FlaskApp	/upload , /cluster , /export endpoints	Orchestrates all components

Key Methods (Pseudocode):

```
class DocumentManager:
    def load_files(files): ...
    def get_documents(): ...
    def export_clusters(assignments): ...

class ClusteringEngine:
    def vectorize(docs): ...
    def cluster(X, linkage): ...
    def get_dendrogram(linkage_matrix): ...
```



3. Data Structure Overview

Data Model	Structure/Fields
Document	{id: str, content: str}
ClusterAssignment	{doc_id: str, cluster_id: int}
LinkageMatrix	Numpy array (n-1, 4) from <code>scipy.cluster.hierarchy</code>

4. Algorithms/Logic

Clustering Flow (Pseudocode):

```
# On /cluster endpoint:
docs = DocumentManager.get_documents()
X = ClusteringEngine.vectorize(docs)
linkage_matrix = ClusteringEngine.cluster(X, linkage_method)
dendrogram = ClusteringEngine.get_dendrogram(linkage_matrix)
assignments = assign_clusters(linkage_matrix, n_clusters)
```

- **Vectorization:** TF-IDF via Scikit-learn
- **Clustering:** Agglomerative (Scikit-learn/Scipy), linkage method selectable
- **Dendrogram:** Generated using Scipy's dendrogram utilities

5. Error Handling

Scenario	Handling Approach
Invalid file format/upload	Return 400 error, show message in UI
Empty or corrupt document	Skip file, log warning, notify user
Clustering failure (bad params)	Return 422 error, show error in UI
Export failure	Return 500 error, log, notify user
Unexpected server error	Return 500 error, generic error message

End of Document