A screenshot of a computer program

Description automatically generated

CSVData Class:

Represents handling of CSV data.

Components:

Attributes:

* filename: string - Stores the name of the CSV file.

Methods:

* \_\_init\_\_(filename: string): Initializes the class with the provided filename.
* read\_data(): list: Reads and parses data from the CSV file.

ElasticsearchClient Class:

Handles interaction with Elasticsearch.

Components:

Attributes:

* connection: object - Represents the connection to Elasticsearch.

Methods:

* \_\_init\_\_(url: string, verify\_certs: boolean, auth: tuple): Establishes a connection with Elasticsearch using the provided URL, SSL verification setting, and authentication details.
* create\_index(index\_name: string, mapping: dict): Creates an index in Elasticsearch with the specified mapping.
* index\_documents(index\_name: string, data: list): Indexes a list of documents into Elasticsearch for a given index.
* search(index\_name: string, query: dict): dict: Executes a search query in Elasticsearch for a specified index and returns the search results.

SearchApp Class:

Acts as the main application orchestrating CSV data ingestion and Elasticsearch interaction.

Components:

Attributes:

* csv\_data: CSVData - An instance of the CSVData class.
* es\_client: ElasticsearchClient - An instance of the ElasticsearchClient class.

Methods:

* \_\_init\_\_(csv\_filename: string, es\_url: string, verify\_certs: boolean, auth: tuple): Initializes the SearchApp with CSV file details and Elasticsearch configuration.
* ingest\_data\_to\_es(index\_name: string, mapping: dict): Ingests data from CSV into Elasticsearch for a specified index with mapping.
* execute\_search(index\_name: string, search\_query: dict): Executes a search query on indexed data in Elasticsearch for a specified index.