* **Name:** Ahmed abdelghany
* **Lab Name:** Cosine Distance Tokenizer Lab

**Application Architecture**

**Overview**

The application is a tokenizer that reads words, assigns unique tokens, and allows retrieval of words and tokens. It uses the Standard Template Library (STL) in C++.

**Software Patterns**

* **Singleton Pattern:** Ensures a single instance of the tokenizer is used throughout the application.
* **Factory Pattern:** Could be used to create different types of tokenizers if the application expands.
* **Observer Pattern:** Useful if you want to notify other components when a new word is added.

**Components**

* **Tokenizer Class:** Manages word-token mapping.
* **Input Handler:** Reads input from a string or file.
* **Output Handler:** Displays tokens and words.

**User-Flow Diagram**

**User Flow**

1. **Start**
   * User initializes the application.
2. **Input Handling**
   * User provides input (string or file).
   * Input Handler processes the input.
3. **Tokenization**
   * Tokenizer reads words.
   * Assigns unique tokens to each word.
4. **Output**
   * User requests token for a word.
   * User requests word for a token.
   * Output Handler displays results.
5. **End**

[Start] --> [Input Handling] --> [Tokenization] --> [Output] --> [End]