**Software Architecture and Design**

**Names: Tommie Walker**

**Esther Nyarangi**

**Group Assignment 1**

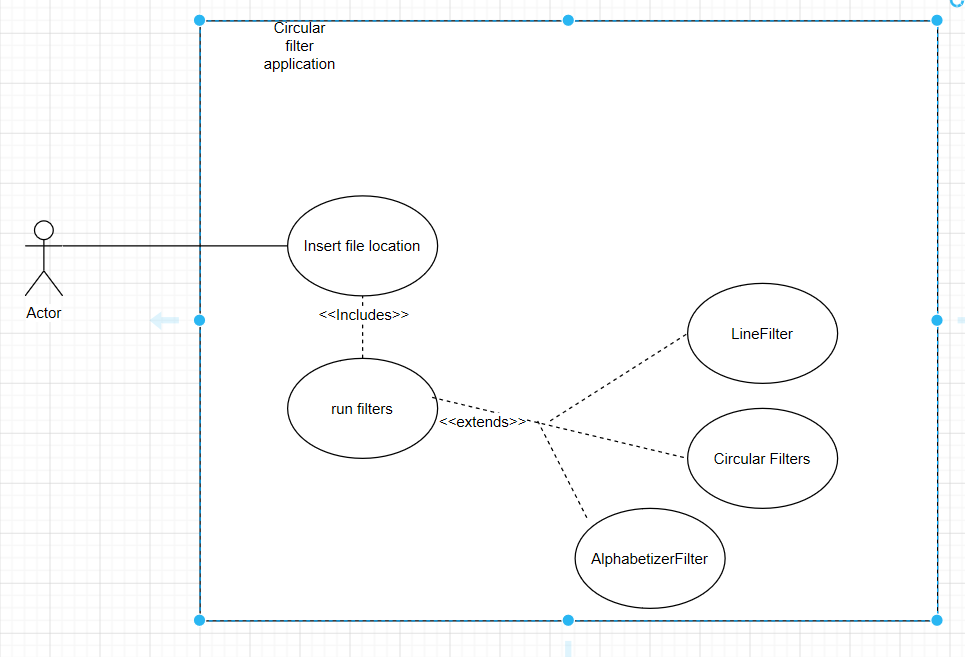
**Requirements Specification**

## **Functional Requirements**

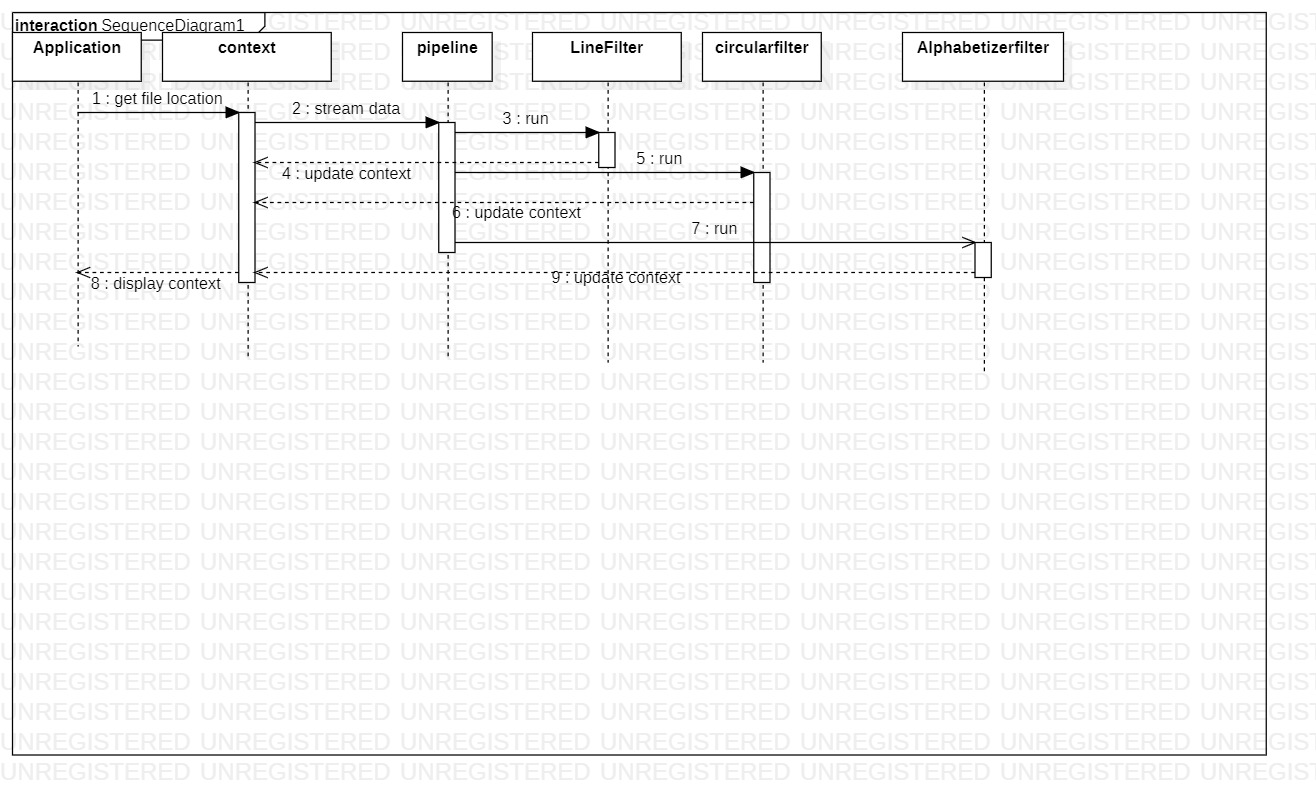
In this case, we define the function of the KWIC system and its components. These functional requirements will drive the choice of software architecture of the KWIC system.

* FR1.0 – The KWIC system shall provide an input field to accept an ordered set of lines.
* FR2.0 – The KWIC system shall accept an ordered set of lines.
  + FR2.1 – Each ordered set of lines are an ordered set of words.
  + FR2.2 – Each ordered set of words are an ordered set of characters.
* FR3.0 – The KWIC system shall perform a circular shift on each the inputted set of lines by repeatedly removing the first word and appending it at the end of the line.
* FR4.0 – The KWIC system shall output a listing of all circular shifts of all ordered set of lines in descending order based off the specified ordered rules below.

**Use Case**



**The Sequence Diagram**



**Architecture Specification**

The pipe and filter architecture are used to implement the system. It is an architectural style/pattern for stream processing. A filter defines a processing or computation step. Data flows through a sequential chain of filters.

The components are the filters whch read the input stream, locally transform data and produce the output stream. We have used three filters:

* Line filter: It gets the input stream and reads it in terms of characters and then passes it to the output stream.
* Circular Shift Filter: It gets the input stream as lines and outputs it after shifting the first word and appending it to the end of the line. It does this repeatedly.
* Alphabetizer filter: It gets the shifted lines and arranges them in an ascending alphabetical order.

The connectors are the pipes. We have used a pipe to move data between the three filters that we have defines above. Those three filters are contained within our Pipe object. When each filter has ran its function, the pipe will return an appropriate output.

**The constraints include:**

* The topology is restricted to linear in pipeline.
* In bounded pipe, the capacity of the pipe is also restricted.
* If more than one filter is pushing /pulling data, synchronization is needed.

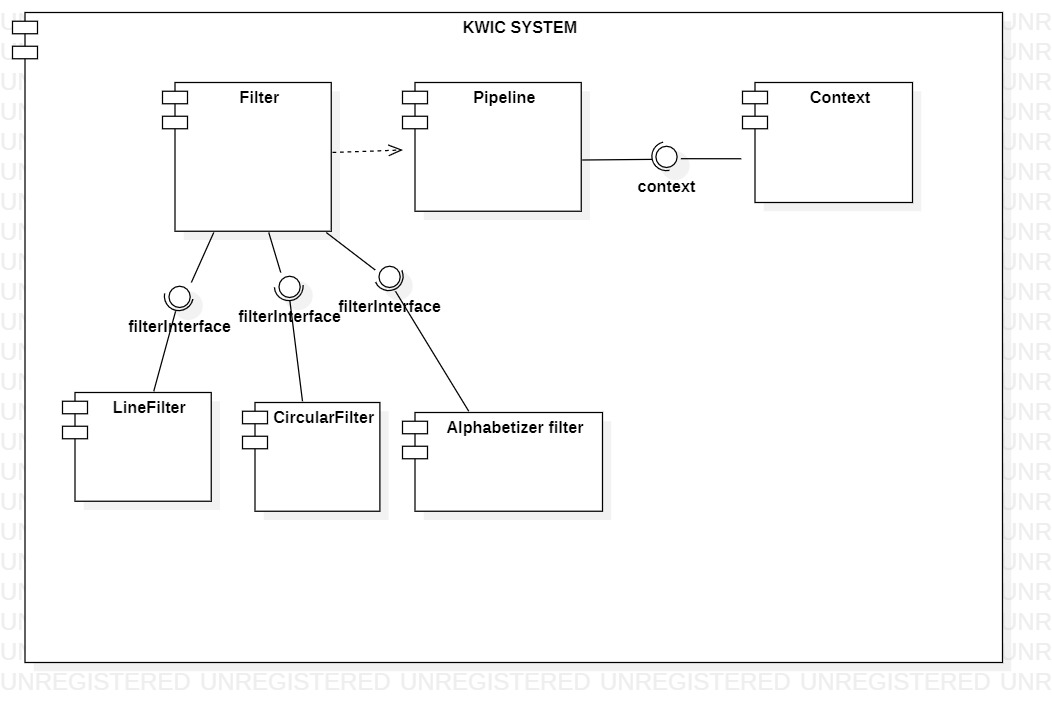
**Advantages of pipes and filters architecture:**

* They support reuse because any two pipes can use whatever order of specified filters to conduct the functions they want.
* Systems can be easily maintained and enhanced because new filters can be added to existing systems and old filters can be replaced by improved ones.
* They support concurrent execution naturally.
* They permit certain kinds of specialized analysis, such as throughput and deadlock analysis.

**Disadvantages of pipe and filter architecture:**

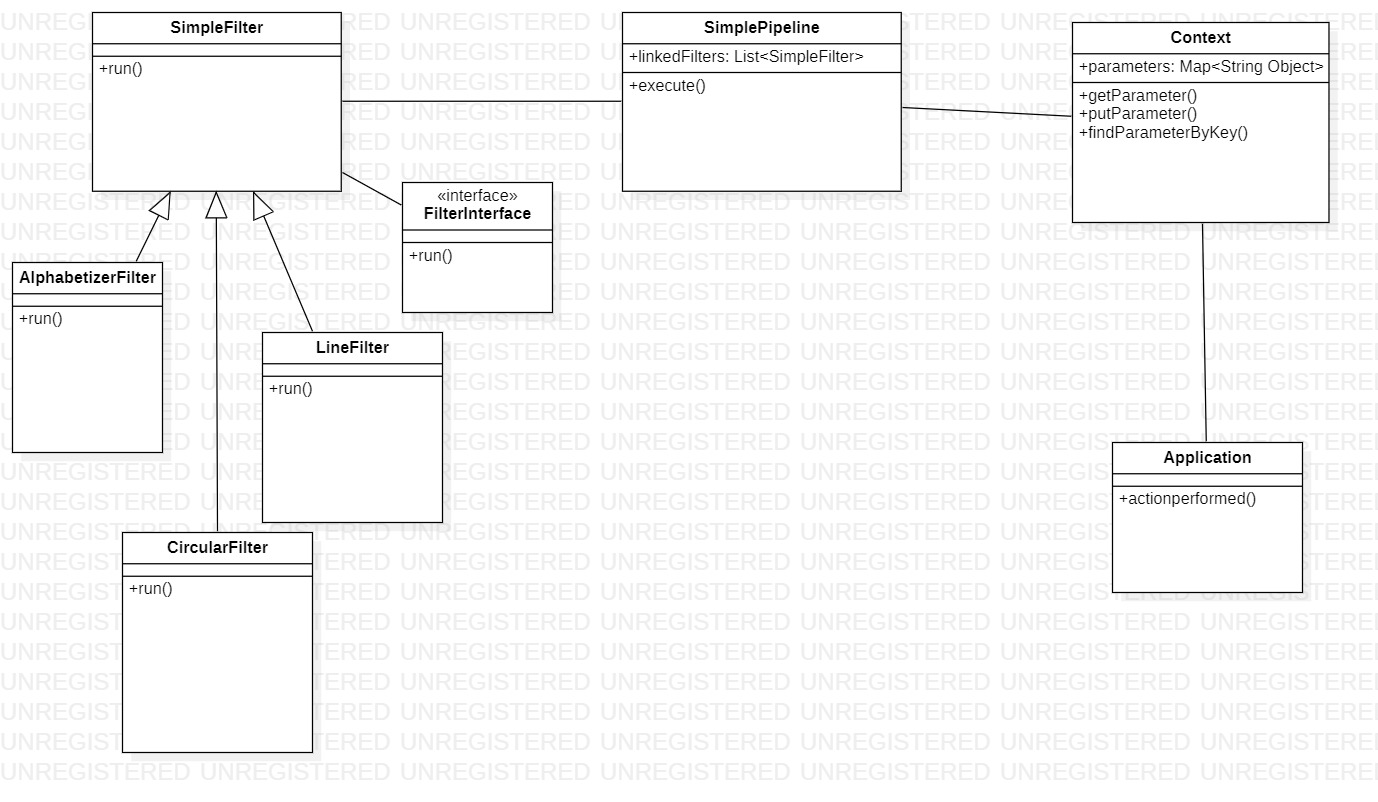
* They have a low fault tolerance threshold.
* They have poor performance because they become complex due to excessive parsing and unparsing.
* They are not good for handling interactive systems.

**The component Diagram:**



**Design Specification:**

**The UML class diagram is as shown below:**



**Implementation specification:**

We have tested our program and it works correctly according to the specified functional requirements.

**User Manual**

Step 1: Enter the Input String .A space is used to separate each line in the input

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

Step 2: Press Start-Index button to start process

A screenshot of a cell phone

Description automatically generated

Step 3: Display of Circular Shifted and Alphabetically Sorted Lines

A screenshot of a cell phone

Description automatically generated