Keyword In Context implementation

Git URL: <https://bitbucket.org/CS3213a1/assignment-1>

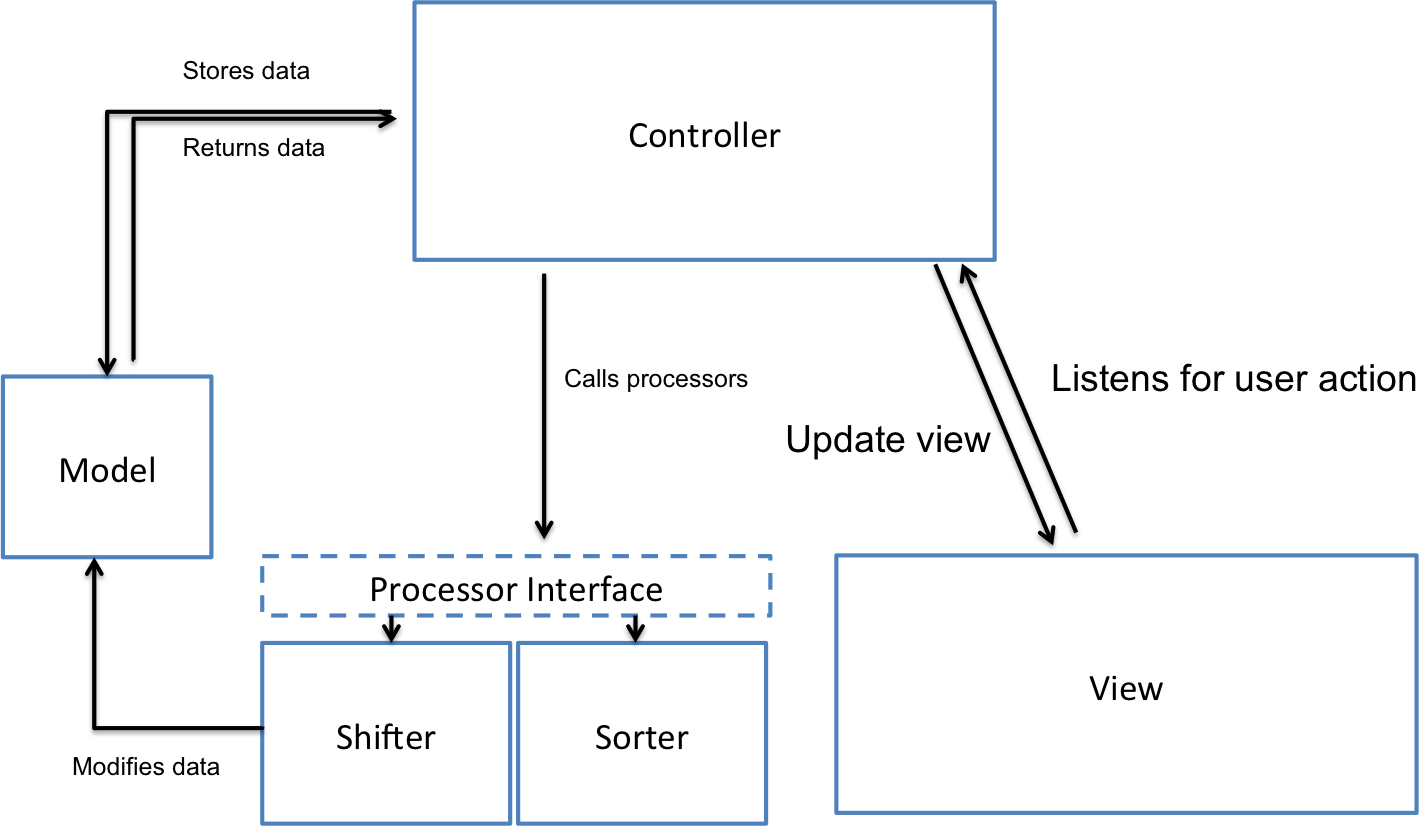
|  |  |  |
| --- | --- | --- |
| Name: | Nguyen Trong Son | Han En Chou |
| Matriculation number: | A0088441B | A0081241U |

# Introduction

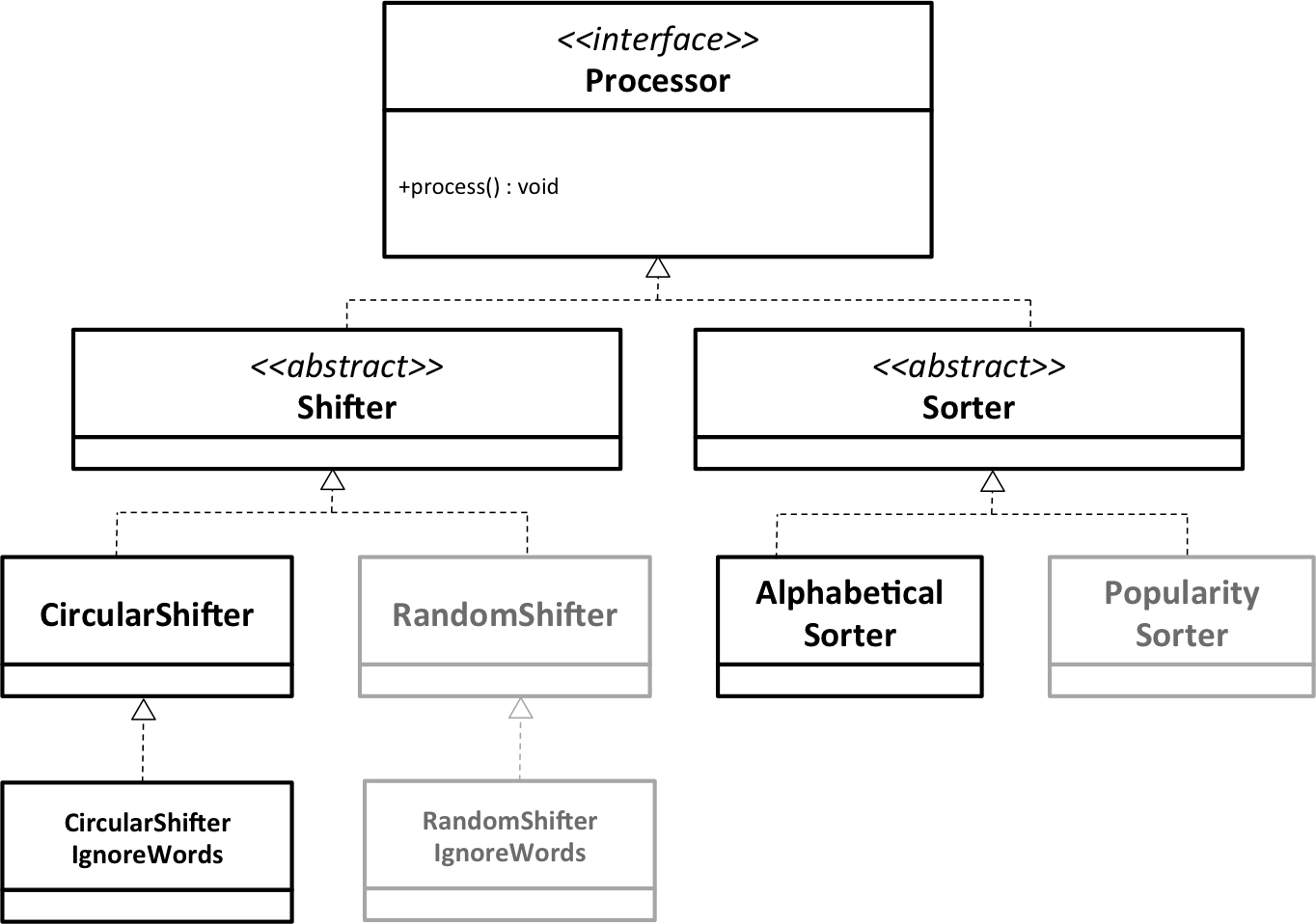
Key Word In Context (KWIC) index system produces a list of permutated text to enable the fast and effective searches based on the keywords. KWIC is widely used for various purposes, such as in searching for movie titles, food recipes and blogs. Due to this wide usage, our design priorities are primarily expandability followed by performance.

# Design Overview

As search engines are front-user facing, it is important to maintain an up-to-date User Interface to manage users expectations. Our implementation has an outline similar to Model-View-Controller, which allows for further enhancement on our implementation. Designers can further improve the User Interface and the model can be extended to implement a logging function for data analysis.



Due to the possible varied usage of KWIC, we have separated the actual processing through a Processor interface. For example, in a movie titles search engine, the sorting may be done using a heap tree based on the recency of the movie. Similarly, the shifting may be done in a randomised manner. This design decision will allow the different types of implementations to optimise the performance of the KWIC for different uses.



# Limitation & Benefits of Selected Design

Our design is Object-Oriented and open to modifications and additions to the processors. It implements a common interface and can be separated in different types of processors (i.e: Shifter and sorter). This allows for the processing units to be understood individually and allows for flexibility. Different developers can work on different modules of the code at the same time.

Our design is based on a batch processing of the lines. We have expedited its performance by processing indices instead of the actual string, and is sufficiently fast for normal usage. However it is a limitation on scalability, as all the lines need to be retrieved from the view component and processed before it is updated again. The controller needs to be modified in order to change the sequence or choice of processors used. The model can be improved to implement a database management system to allow for persistent data and concurrent access.