**CHAPTER 8**

**DETAILED DESIGN**

**8.1 Application architecture**

The Input 🡪 Process 🡪 Output model is followed for each of the following stages:

* Stylistic evaluation and generation of comments
* Display of comments
* Text highlighting
* Adherence to templates
* Conformity to target readability
* Opening an existing file
* Saving an edited file
  + 1. **Stylistic evaluation and generation of comments**

Input Process Output

**Text for evaluation**

**Evaluate against rules of style**

**Comments**

***Fig 8.1.1*** *Generation of comments*

The text is taken for evaluation against a predefined set of rules of style. Comments are generated for every violation of rules of style. Each comment has a priority depending on the importance of that rule of style.

* + 1. **Display of comments**

Input Process Output

**Comments**

**Display of comments as in list in GUI**

**Colour-coded comment list**

***Fig 8.1.2*** *Display of comments*

The comments generated after evaluation of the text should be displayed to the user. They appear as a list. The colour of each comment depends on the priority assigned to it. The user can view comments of only one priority or all comments.

* + 1. **Text highlighting**

Input Process Output

**Clicked comment**

**Text search**

**Highlighting of text**

***Fig 8.1.3*** *Text highlighting*

When the user clicks on a comment, the text to which the comment refers gets highlighted. Even if the text is not visible on the screen, the file scrolls down so that the text is visible when it is highlighted.

* + 1. **Adherence to templates**

Input Process Output

**Template option**

**Displaying the template for user to insert text**

**Text in editor window**

***Fig 8.1.4*** *Adherence to templates*

When the user chooses a template to follow, the template appears along with boxes to the user to enter textual data. Upon completion of this action, the text along with help text appears in the editor window. The user can further edit this.

* + 1. **Conformity to target readability**

Input Process Output

**Target readability**

**Evaluation of readability**

**Readability highlighting**

***Fig. 8.1.5*** *Conformity to target readability*

The user selects a target readability score. The next time the user requests for readability analysis, the results are highlighted if the text does not lie in the target readability range. In case of auto evaluation of readability, the results are highlighted as and when the user types in the editor window.

* + 1. **Opening an existing file**

Input Process Output

**File name**

**Opening file**

**File contents in editor window**

***Fig.8.1.6*** *Opening an existing file*

If the user wishes to open an existing file he may search in the system directory and chose a file. That file is read and its contents are visible in the editor window.

* + 1. **Saving an edited file**

Input Process Output

**Text**

**Transferring contents to a file**

**File with text**

***Fig 8.1.7*** *Saving an edited file*

When the user wishes to save his work, if the file already exists, then it’s contents are replaced by the current text in editor window. However, if the file does not exist, then the user must choose a file name and the directory in which he wishes to save the file. The file data can now be accessed outside the application.

**8.2 Modules**

Modules are major components of design that may translate to a function or a class in code. Enumeration of modules makes the further implementation phase easier.

**8.2.1 Parsing of sentences**

The user types his document in the editor window or opens an existing document. Once he is satisfied with his work, he can ask for his document to be evaluated stylistically. After reading of the text, it is split according to sentences, then words and finally each word is tagged with its appropriate part of speech.

Return tags

End

Tag resulting words

Start

Read text

Split text per ‘\n’

Split result into sentences

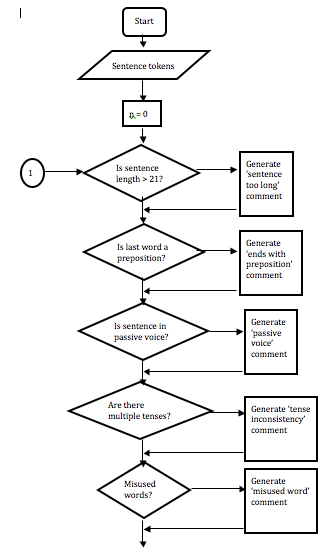
Split result into words

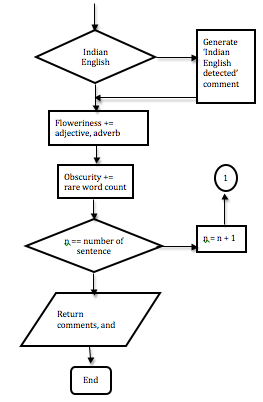
***Fig.8.2.1*** *Sentence tokenization and tagging*

**8.2.2** **Generation of comments**

Once the text is parsed and each word tagged with the appropriate part of speech, comment generation occurs. The comments refer to non-conformance of a sentence to a rule of style. The following types of comments are generated:

* Length of a sentence: The number of words for a moderate sentence length is 21. Beyond this number, the sentence is considered too long.
* Ending with a preposition: Ending a sentence with a preposition is not considered a good practice. Non-conformance to this rule gives rise to an error.
* Usage of passive voice: Using passive voice results in an indirect main object. This makes the sentence weak and should not be used frequently.
* Tense inconsistency: In sentences sticking to a single tense gives clarity of writing. It removes a possibility of confusion.
* Misused words: Several words in English are similar sounding. For example: effect and affect, then and than. One can be mistakenly used for the other.
* Usage of Indian English: Several phrases have been accommodated into Indian English and the original/correct phrase is usually unknown. So, such phrases are detected with alternative phrase suggestions.
* Floweriness: It is the number of descriptive words per sentence. Descriptive words refer to adverbs and adjectives.
* Obscurity: It is the number of rare or infrequent words of English per sentence.





***Fig.8.2.2*** *Generation of comments*

**8.2.3 Template adherence**

If the user wishes to create a document of a particular template he may choose one from the options given. The templates provided are:

8.2.3.1 School essay

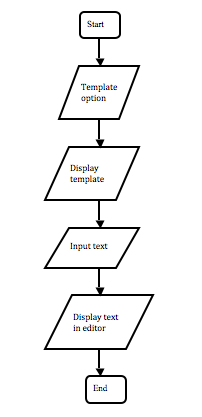
* Title of the essay: The title must be concise and relevant to the essay
* Introductory paragraph
* First point considered
* Second point considered
* Third point considered

8.2.3.2 Debate

* Topic of debate
* Introduction to topic: Specify whether for or against the topic
* Argument one
* Argument two
* Argument three

8.2.3.3 Formal letter

* Sender’s address: Limit address to 3 lines
* Receiver’s address
* Date
* Subject
* Salutation
* Contents of letter
* Regards …



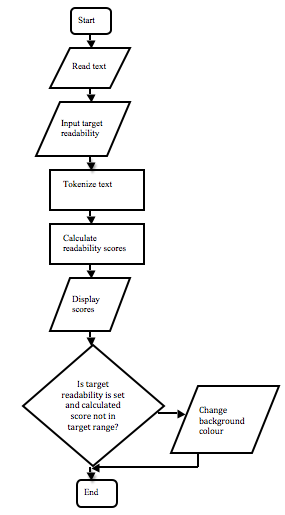
***Fig.8.2.3*** *Templates*

**8.2.4 Readability score**

If the user wishes to obtain readability scores for his document, then the application extracts the text, evaluates against various readability scores. These scores are displayed. If the user has specified a target readability range, then when the readability score is calculated the background colour denotes whether it is within the range or not.

The following readability measures are considered:

* Flesch Reading Ease: Range is from 0 to 100
* Flesch-Kincaid Grade Level: Most widely used score
* Gunning-Fog Score: Considers complex words (i.e. words with 3 or more syllables)
* Coleman-Liau Index: Measures average number of letters and sentences
* SMOG Index: Considers number of polysyllabic words
* Automated Readability Index: Considers easy-to-parse parameters like word count and character count.

****

***Fig.8.2.3*** *Readability score*

**8.3 Database Design**

The number and type of comments are fixed in the application. Another candidate for usage of databases is misused words/phrases. But the number of misused words/phrases is fixed and more will not be added during program execution. Hence there is no reason to bring in databases.

However, data must still be stored. Misused words/phrases are stored in the form of a list of tuples.

Misused words:

[(‘word1’, ‘tag1’, ‘replacementWord1’),

(‘word2’, ‘tag2’, ‘replacementWord2’),

…

(‘wordN’, ‘tagN’, ‘replacementWordN’)]

Misused phrases:

[(‘misusedPhrase1’, ‘correctPhrase1’),

(‘misusedPhrase1’, ‘correctPhrase1’),

…

(‘misusedPhraseN’, ‘correctPhraseN’)]