**CHAPTER 4**

**PROJECT REQUIREMENTS DEFINITION**

**4.1 Project Perspective**

The user in order to use the application must start with either a blank document or open an existing file in the editor window. The text in the window can then be evaluated for its floweriness, obscurity and readability.

In addition to that, numerous rules of style are checked. The rules are a combination from the book “The Elements of Style” by Strunk and White and some universally accepted guidelines for writing appropriately. These are displayed in forms of suggestions and comments to the user which he/she can choose to ignore or act upon.

Some of the rules include an appropriate length of a sentence, discouraging use of passive voice and ending a sentence with a preposition, tense inconsistency in a sentence, etc.

It also includes features for detection/correction of commonly misused words in English, and some misused phrases in Indian English. The comments for these also suggests alternatives for the correction of these phrases

**4.2 Project Functions**

The major facilities that are provided by our system are the following:

* Evaluation of readability of a document
* Templates for various formats such as Debate, School Essay and Formal Letter
* Evaluation of the floweriness of the document
* Evaluation of the obscurity of the document
* Detecting very long sentences
* Detecting the use of passive voice
* Detecting and providing alternative to misused words
* Detecting and providing alternative to misused phrases in Indian English
* Detecting tense inconsistency
* Checking if a sentence ends with a preposition
* Typing text into the editor window, with advanced features like copy-pasting and cut-pasting
* Checking if the readability matches the target readability or not
* Creation, Opening and saving a text file

**4.3 User Classes and Characteristics**

The user can open a text file in the editor window or type directly and save the file later. They can choose the purpose of the document and adhere to the template provided for the same.

They can get their readability scores by clicking on Calculate Readability option under Tools menu or by pressing Ctrl+G. They can also choose readability to be evaluated dynamically by clicking the Auto-evaluate readability option under the Tools menu.

They can choose to evaluate the text against the set rules of style by clicking the Evaluate Readability option under Tools menu. When they click on the comment, the sentence for which the comment was generated gets highlighted. The comments are displayed in different colours based on their severity. The user can choose which set of comments to display by clicking the toggle buttons above the comments section.

* 1. **Operating Environment**

4.4.1 Hardware requirements

* Processor : Intel Core i3 and above
* RAM : 2GB and above
* Hard disk : 20GB and above

4.4.2 Software requirements

* Operating System : 64 bit Windows or linux
* Language : Python 2.7
* Python packages : NLTK
* UI: PyQT

**4.5 Assumptions and Dependencies**

The biggest constraints for this project are the limitations of the Natural Language processing subsystem, in this case NLTK. The system is only as good as the parser it uses; hence any wrong tagging/tokenizing/parsing on NLTK’s front shall also result in wrong output in the system. An assumption for this system is that the user is familiar with the English language.

.