|  |  |
| --- | --- |
| Requirement 1 | Must be able to detect basic parts of speech |

This requirement is handled through our use of dictionary files. Early on in our development we acquired a massive list of English words, arranged alphabetically and tagged with a character indicating any parts of speech that any one word could be. We wrote a parser and sorted this large file into separate files containing nouns, verbs, prepositions, and articles. In our WordProcessor class the user’s input is split into individual words. Each word is sent to our WordTagger class which attempts to find it in these files. If found, the program will wrap the word in the appropriate PartOfSpeech object. If the word is not found the program will indicate that this is the case and the word can be manually added to the appropriate file.

|  |  |
| --- | --- |
| Requirement 2 | Must be able recognize a valid sentence as defined by our grammar rules |

This requirement is handled by the languageProcessor class which uses a basic natural language processing algorithm. According to our grammar rules a valid sentence is composed of a noun phrase and a verb phrase. Noun phrases and verb phrases are composed of various, grammatically correct, combinations of nouns, articles, verbs, and prepositions. The algorithm is able to discern proper grammar by knowing when and how to perform these combinations.

|  |  |
| --- | --- |
| Requirement 3 | Must be able to interact through the console |

This requirement is met by our ConsoleInteraction class. This class is able to obtain input from the user and print output statements to the user.

|  |  |
| --- | --- |
| Requirement 4 | Must Respond with relevant statements or questions |

This requirement is handled in our WordProcessor class. This class contains methods that are able to construct replies out of valid sentences. These replies are built out of a selection of pre-built response “templates”. The methods are able to grab relevant parts of the valid Sentence object and place them into the templates which are then printed to the user. The methods will also attempt to conjugate and or pluralize nouns and verbs in order to produce grammatically correct replies. However, this is not always successful due to the complexity of English verb conjugation. On the bright side the output is often hilarious.

|  |  |
| --- | --- |
| Requirement 5 | Must be able to tell the user when it is unable to understand the input |

This requirement is handled in two places. First, in the WordTagger class, if the program is unable to find an input word in any of the dictionary files it will alert the user that it was unable to understand as it technically doesn’t “know” the word. Second, in the event that each word in the users input is tagged appropriately but the LanguageProcessor class is unable to combine these words into a valid sentence the LanguageProcessor class will return null. When the word processor receives detects this null it will then prompt the user that it was unable to understand the input.