**Requirements Acquisition Tool**

**Group 6 - Team**

**Wenbo Hu**

**Prajwal Shet**

**Nabilahmed Patel**

**Limegh Shen**

**Sindhu Velli**

**Tejaswini Mariyada**

**Project Description**

The Requirement Acquisition Tool is a JAVA based software tool which converts the input given by the client in the form of a textual document into software requirements for the software engineers to work with. The tool uses predefined rules, which are applied on the textual document to get the required output. The GUI is divided into three parts: Input text-area for the clients to open, edit and modify the text, the right panel which consists of buttons for opening, processing, generating and saving documents and the output area which displays the list of requirements.

The following are the functional requirements:

R.1. The RAT must be programmed by JAVA.

R.2. The RAT must be platform independent.

R.3. The RAT must provide a text-area for the user to provide the input.

R.3.1. RAT must be able to import previously saved text documents.

R.3.2. RAT must allow user to type in the requirements and edit them.

R.4. When the user clicks on the “Process” button, RAT must be able to implement Natural Language Processing on the input text.

R.4.1. RAT must be able to highlight the transitive verbs and nouns in the input text.

R.5. The User must be able to edit the highlighted text after the NLP incase the highlighted text is not domain specific.

R.6. When the user clicks on the “Generate” button, RAT must be able to generate list of requirements.

R.6.1. RAT must take the highlighted nouns and verbs as inputs and identify other domain specific rules including “X of Y” expressions, adjectives, numeric, possession expressions, “constituents/part of” expressions, “containment/containing” expressions and “X is a Y” expressions.

R.6.2. RAT must be able to form sentences using the identified words following set of pre-defined rules.

R.7. RAT must have a “Save” functionality, which allows the user to save the output in a file.

R.8. RAT must use sound software engineering principles for its development and deployment.

R.9. RAT must undergo unit testing, integration testing and acceptance testing and the test cases must be documented.