**Stage 1 - Project Proposal**

| **Team Members:** | Ayan Hazarika, Sahil Basra | **Marks**  **(for teacher-use only)** |
| --- | --- | --- |
| **Software Project:** | Text Analysis Software |  |
| **Description:**  Describe the program, including who the program is intended for, rules and/or instructions explaining how the program functions, etc. | ***What is the program, and how does it work?***  This program accepts a text file, analyzes the data within the file and converts it to a graph - which is then used to determine word frequencies, predict most probable sentences (similar to autocomplete), provide sample sentences/paragraphs, compare writing styles to other pieces of texts, alongside many other features. The intended target audience for the program is for people in the field of stylometry, and any individual who wishes to analyze their own writing. When the program executes, it will initially display an in-depth description to the user of the program’s features and functionalities. | **/2** |
| **Details:**  Briefly explain some of the programming structures you will use to complete the program. | This program utilizes a variety of programming concepts from this course, including:   * **Nodes and Graphs:** each node will represent an individual word, and a graph will be used to represent related words (i.e. words that appear in conjunction with one another) * **Depth First Search:** will be used to traverse the graph when searching for a specific word * **Adjacency Matrices & 2D Arrays:** will be used to store frequencies between words (i.e how often certain words repeat after another) * **Binary Search:** used to search through the arrayList of nodes * **Recursion:** will be used for DFS and Binary Search * **Objects, Classes:** graph class, node class, word object, sentence object… * **Arraylist (nodes):** will be used as a cache to store nodes * **Private, Protected and Public Data Properties:** some attributes that do not need to be accessed by the client will be stored as private data | **/2** |
| **IPO Chart:** |  | **/6** |

| **Input**   * What information is needed from the user? * What variables will be needed to store that information? Provide names and types. * How will you make your input user friendly? | **Processing**   * What needs to happen to the data before it can be output? * What formulas are needed? | **Output**   * What needs to be output to the user? * How will the output be formatted? * How will you make your output user-friendly? |
| --- | --- | --- |
| ***What information is needed from the user?***   * Text File(s)   + One file will contain the data to be analyzed, another file will contain the data to be compared to * Option Choice of which feature they would like to execute   ***What variables will be needed to store that information? Provide names and types.***   * *ArrayList* of node objects * *String* variable containing word * *2D Array* adjacency matrix containing frequencies   ***How will you make your input user friendly?***   * When the program executes, a menu will be displayed explaining the entire function of the program, and also presenting numbered choices for the different possible features * The user will also be informed if there are issues with the text file (FileNotFoundException, issues with the text file’s name…) | ***What needs to happen to the data before it can be output?***   * The text paragraph(s) will be converted into graphs where the vertices (nodes) represent individual words, and the edges represent frequencies between words. An adjacency matrix   *An explanation for each of these features can be found below:*   * *Word Frequency:* will search the node arraylist for the most frequently occurring word * *Most Probable Sentence:* uses the adjacency matrix and edges between nodes to predict the most probable sentence (i.e. works like autocomplete on your phone) * *Possible Next Sentence/Paragraph:* analyzes the user’s writing style (frequently combined words, most common transition and intro words…) and creates a sample next paragraph for the user * *Comparing Texts to determine similarities in writing style:* will compare the frequency of words which appear in both texts * *Average Sentence Length:* searches through the arrayList of nodes and averages all the sentence lengths * *Average Word Length:* searches through the arrayList of nodes and averages all the word lengths   ***What formulas are needed?***  No formulas are used for our program | ***What needs to be output to the user?***   * Whichever option they chose at the beginning of the program; Most Probable Next Sentence, Sample Next Sentence, Sample Next Paragraph, Word frequency, Comparing and contrasting tests…   ***How will the output be formatted? How will you make your output user-friendly?***   * The results of the analysis will be presented to the user in simple text format. In order to make the results more readable and interesting, varying colors will be used. |

**Schedule:**

Fill in the calendar indicating completion goals and team member responsibilities.

| **Sunday** | **Monday** | **Tuesday** | **Wednesday** | **Thursday** | **Friday** | **Saturday** |
| --- | --- | --- | --- | --- | --- | --- |
| 4 | **5**  Culminating Projects Introduced | **6**  Final Unit Test | **7**  In Class Coding Begins | **8**  ***Brainstorming Project Ideas*** | **9**  *Sahil and Ayan away for prom* | 10 |
| 11 | **12**  ***Finalize Project Idea*** | **13**  ***CODING:*** Begin Graph and Word Class  ***WRITTEN:*** Work On Proposal | **14**  ***CODING:*** Finish Graph and Word Class  ***WRITTEN:*** Submit Proposal, Start Program Design | **15**  ***CODING:*** Begin Coding Features  ***WRITTEN:*** Finish and Submit Program Design | **16**  ***WORK FROM HOME:***  Continue Coding Features | 17  ***WORK FROM HOME:***  Finish Coding Features |
| 18  ***WORK FROM HOME:***  Final Touches (Error Proofing, Documentation…) | **19**  Culminating Projects Due  Reflections Completed in Class | **20**  Period 4  Grade 12 Exams |  |  |  |  |