**Part B: Documentation (50%)**

Documentation supporting your software development journey must be created using Microsoft Word and include the following sections:

|  |  |
| --- | --- |
| Section | Description |
| Planning | Task Definition |
| Structure charts |
| Flow chart |
| Data Dictionary |
| Implementation | Link to your GitHub repository containing source code |
| Testing | Test table |
| Evaluation | Project reflection |

## Task Definition

I have been assigned the task of developing ‘connections.py’, a word puzzle driven python application for the 2023 game of connections, ensuring a functional and fun game play that reflects the game's challenging nature. I aim to execute a will programmed, bug-free application that is user-friendly and more importantly engaging to the user.

The functional requirements I will need to implement include:

* Randomly select 4 categories and in each of those categories it will select there 4 words.
* Display those 12 words in a 4x4 words grid.
* There will be 4 lives.
* For every wrong guess the live will deplete by 1 and for every right answer it will display how many live you have left and congratulate you for guessing all the words correctly with their categories.
* If all your live are less, then 1 so 0 it will display game over.
* Have a restart game function.

## Structure Chart

## Algorithm Design

**Flow Chart**

## Data Dictionary (Fix the Table)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Variable | Data Type | Format for display | Size in bytes | Size for display | Description | Example | Validation |
| word\_list | List[String] | List of strings | - | - | List of potential secret words for the game | ["apple", "banana"] | Must not be empty |
| secret\_word | String | Text | - | - | The word to be guessed by the player | "apple" | Must be from word\_list |
| guessed\_letters | Set[Char] | Set of characters | - | - | The set of letters that have been guessed | {'a', 'e'} | Unique characters only |
| attempts | Integer | Numeric | 4 | 1-2 digits | Number of attempts left for incorrect guesses | 6 | 0 to max number of attempts |
| game\_won | Boolean | True/False | 1 | True/False | Flag to determine if the game has been won | True/False | True or False only |
| guess | Char | Single character | 1 | 1 character | Current letter guessed by the player | 'a' | Single character |