**Consider FURPS:**

Functional Requirements

Usability Requirements

Reliability Requirements

Performance Requirements

Supportability Requirements

**Hangman Requirements:**

|  |
| --- |
| **Functional Requirements: What must the App need to do? What are the features?** |
| The app must select a word randomly from a collection of words. |
| The app must be able to show the player how many letters are in the word that the player is guessing |
| When a player guesses a word letter, the app must determine if the letter is in the word |
| If the letter is in the word, then the app must be able to place the letter at the appropriate location in the word. |
| If the letter appears more than once in the word, the app must show all occurrences of the word. |
| The app must be able to display these results to the player. |
| The app needs to handle incorrect player-guesses |
| The app needs to determine when the player loses the game. Limit incorrect guesses  It needs to display the consequence of the players incorrect guess using the Hangman rules. |
| The app must determine when the player wins. |
| The app should register number of moves required to win. |
| **I DON’T KNOW WHAT MAIN METHOD WILL DO YET – HOW WILL THE APP RUN/?????????????** |

|  |
| --- |
| **Non-Functional Requirements: Ex Help documentation, support, runtime efficiency** |
| Ex: System must be able to perform searches within 2 secs. |
| The system must be able to read 26 csv files. Each file contains a list of words. Each contains words from a single letter of the alphabet. |
| The app needs to write to a file, the player and the player profile. [ words used, wins, loses, ] |
|  |

|  |
| --- |
| **Application Description: Describe in plain English** |
| The app will model the word game Hangman. The app will have interaction between a host, who knows |
| The secret word and a player who will be trying to guess the word letter by letter. |
|  |
|  |
|  |

|  |
| --- |
| **Identify Main Objects** |
| A Wordbank object - |
| A Host Object - |
| **Player :** has an ID, a name, access to his/her record, |
| **Player:** |
| **Administrator:** This object will manage reading and writing player data to and from a file.  Data – Like – words that this player has won, lost, how many turns it has taken for a win  Lookup players to let host know if this player is new or revisiting the application. |

|  |
| --- |
| **Describe object interactions/relationships** |
| **Host**: needs to select Word |
| **Host**: needs to analyze word |
| **Host**: stands between Word and Player - |
| **Host**: hides word from player |
| **Wordbank** : selects a word when asked |
| **Wordbank:** knows how many words are in the bank |
| **Wordbank:** keeps track of words that have been used within a session. [ need to store selected words in some kind of container so that it does not get re-used in the same session – maybe even with the same player. |

|  |
| --- |
| **Create a Class Diagram** |
|  |
|  |
|  |
|  |