## Hangman Functional Requirements:

1. Starting the Application

If a word is obtained from the Word Server, game play begins by offering the player their first turn.

B. Playing a game

At each turn the application will display a visual indicator of how many letters are in the hidden word and if any of the letters have been correctly guessed they are shown in the proper position in which they appear in the word.

The application will display a "guess count" which shows how many incorrect guesses the player has made.  An incorrect guess is guessing a letter which is not in the hidden word.

The application will allow the player to enter a letter.

If the letter entered is not between A and Z display a message "Invalid move" and allow the player to guess again (without penalty).

When the player enters a valid letter the application will check to see if the game is over and if not will continue to the next turn.

C. Ending a game

The player wins by correctly guessing all the letters in the hidden word.

The player loses if he/she makes seven incorrect guesses.

If the player wins the application will display a congratulatory message.

If the player loses the application will display a consolation message and will reveal the hidden word.

When the game is over (either win or loss) the application will offer the player an opportunity to begin another game.

If the player indicates they want to play again, a new game is started.

If the player indicates they do not want to play again, the application is terminated.

Note: there is no way for a player to request termination of play during the middle of a game.

## Hangman Non-Functional Requirements

## General Guidelines:

## Performance and reliability are not very important.  Priority  should be given to adaptability, maintainability, and usability.

Modifiability

1. If it is desired to change the number of turns in a game, the developer will be able to make the required changes in < 1 person-hours.

Adaptability

1. The user must be able to specify an alternate user interface at execution time on the command line.
2. Any alternate user interface must be able to "plug in" at run time without recompiling the application.

Reliability

1. Since the program is purely for recreation and involves no user data, reliability is of low importance.

Security

1. The program will not access any user data files or programs.
2. The program will not alter or replace any system files.

Usability

* A new user should be able to play a complete game of hangman in less than ten minutes.
* A new user should commit less than one error in use of the game (e.g. selecting the wrong letter) every ten minutes
* A user who is familiar with the rules of Hangman be able to correctly operate the program without any written documentation.

Performance

1. Desired Response Times (not critical) : At game start: less than five seconds

Maintainability

1. It is desirable that the game administrator be able to modify the words data file using a simple text editor.

**External Interfaces**

The Best Care Medical Store Management System will use the standard input/output devices for a personal computer. This includes the following:

* Keyboard
* Mouse
* Monitor

**Software Interfaces**

* Operating System: Windows 10
* IDE: Code Blocks
* Language: C

**Communication Interfaces**

This system shall be a stand-alone product that does not require any communication interfaces.