

# ELEC1005 PROJECT 1 REPORT

## Group Members

- Rohith Nedunchezian - 510226668 - Planning
- Inika Anand - 510565604 - Implementation
- Noe Chacko Jacob - 510207807 - Implementation
- Ryan Padamadan - 510283373 - Implementation
- Joshua Chao-Hsu Chen - 510656681- Quality Assurance
- Ryan Cheung - 520103148 - Planning

## 2. Planning

The main object of this project is to improve the given template game: Hangman as a source of entertainment. This objective is achieved by building on the project's user interface and experience and adding additional features to enhance the game's functionality.

### 2.1 Requirement analysis

The user requires this game to be a source of entertainment, hence the game should have features that appeal to the user such as a good user interface and experience. Additionally, the game should create an attractive environment whereby the user wishes to continue playing to compete with other players.

Based on the main objective of this project, the requirements are split into two types: Architectural & Design requirements and System & Implementation Requirements.

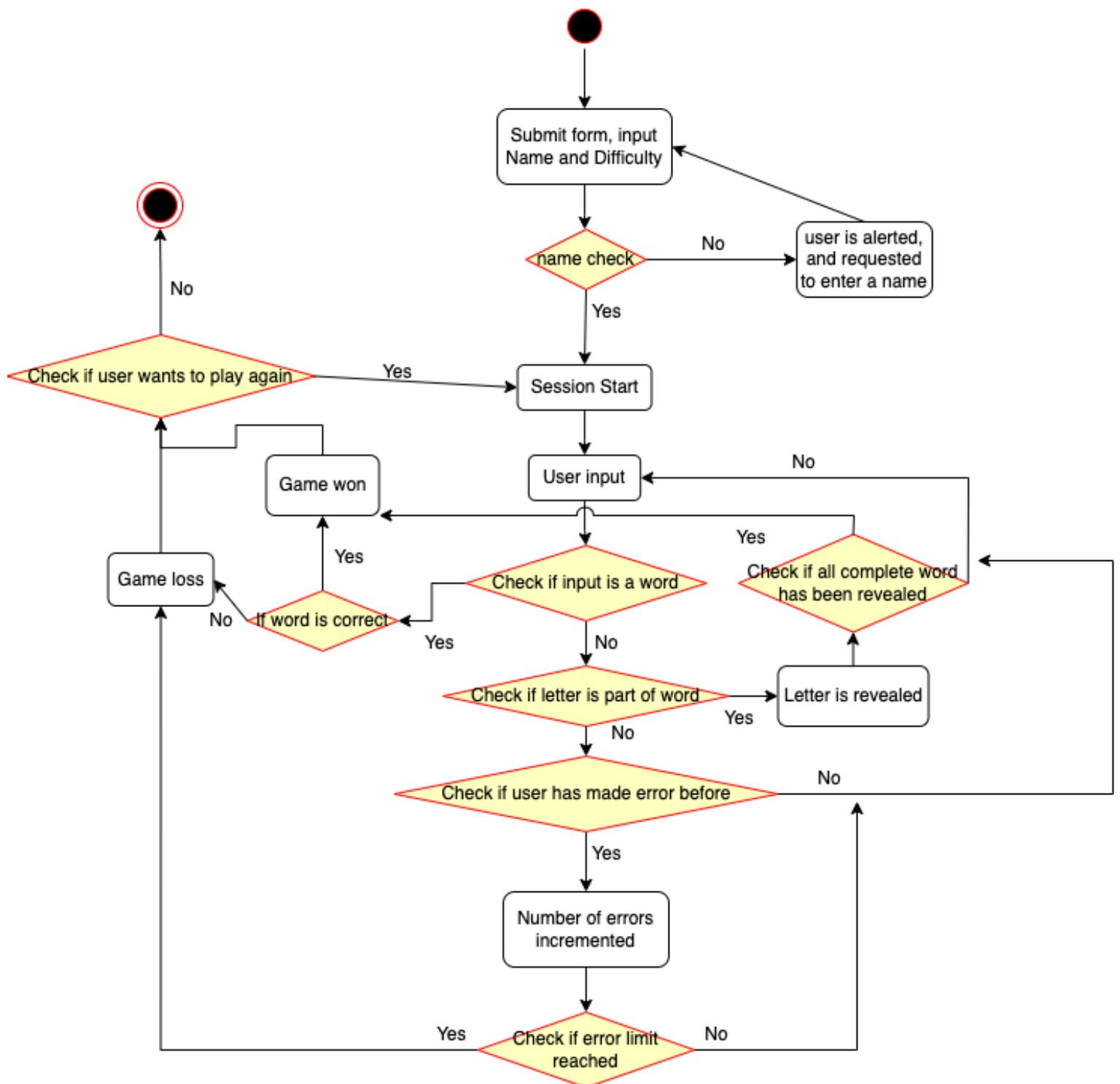
Architectural & Design requirements focus on the game's front end, which tackles the user interface and experience. The software achieves this by adding background images, typography, and a colour scheme while maintaining a balanced and uncluttered layout. The software improves user experience by creating popup alerts, hoverable features, and sound effects.

System and implementation requirements contain all the backend and additional features that have been added to the game. Difficulty levels were added to the game to categorise each game on the leaderboard, this builds a competitive environment for the players. The user can choose to input a single letter or the whole word at a risk, which is an additional feature that was added to the software to make it easier for the user to

finish the game. If the user inputs the correct word then the user wins, otherwise, the user loses the game instantly. Lastly, errors for each letter entered are only counted when players make at least one correct guess. This allows the user to play a fair game where the user can only guess wrong letters after guessing the first right letter.

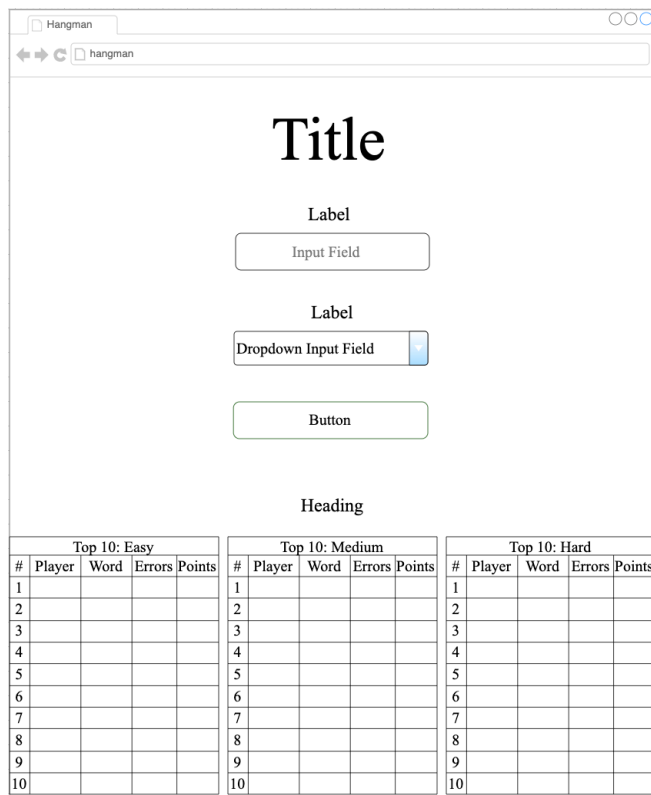
## 2.2 UML (Activity Diagrams)

The following is a UML diagram for the entire hangman game.



## 2.3 Wireframe

The following contains the wireframes for each page in the website.



Wireframe for Page 1: Home Page. The page features a browser window with the title 'Hangman' and a URL 'hangman'. The main content area includes a large 'Title' placeholder, a 'Label' above an 'Input Field', another 'Label' above a 'Dropdown Input Field', and a 'Button'. Below these elements is a 'Heading' and three tables displaying the 'Top 10' players for 'Easy', 'Medium', and 'Hard' difficulty levels. Each table has columns for '#', 'Player', 'Word', 'Errors', and 'Points'.

#	Player	Word	Errors	Points
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

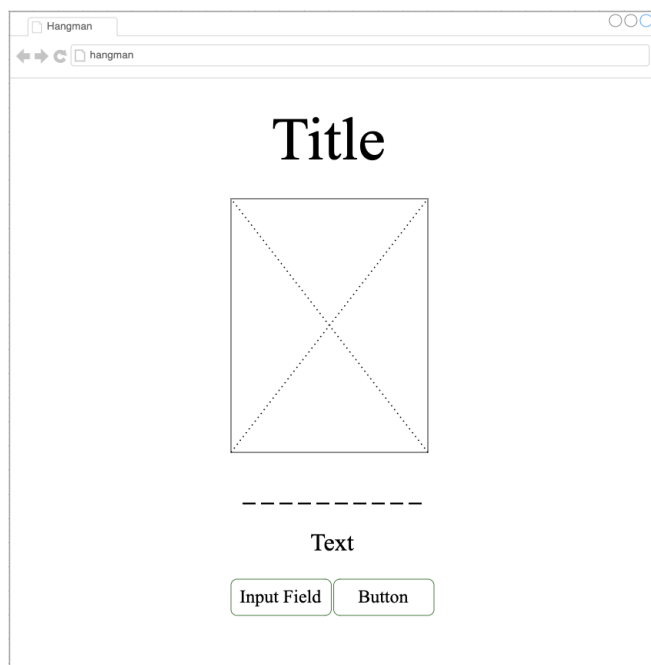
#	Player	Word	Errors	Points
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

#	Player	Word	Errors	Points
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

### Page 1: Home Page

Elements:

- Title → Hoverable feature which changes color upon hover.
- Input Field (Datatype: String) → for player name with label to understand input requirements.
- Dropdown Input Field (Datatype: String) → selection options available between easy, medium and hard. Default selection is set to easy.
- Button (Type: Submit) → Begins game and submits form content. Hoverable feature that adds background image to button upon hover.
- Table → Displays top 10 players. There are three tables for the categories: easy, medium and hard.



Wireframe for Page 2: In Game Session. The page features a browser window with the title 'Hangman' and a URL 'hangman'. The main content area includes a large 'Title' placeholder, a square shape with a dashed 'X' inside, a dashed line, and a 'Text' label. Below these elements are an 'Input Field' and a 'Button'.

### Page 2: In Game Session

Elements:

- Title → Hoverable feature which changes color upon hover.
- Shape → HTML shape that appear based on number of errors
- Hidden word → Will display letters inputted
- Text → Displays number of errors
- Input Field (Datatype: String) → for entering a letter or the entire word.
- Button (Type: Submit) → Submits the letter or word entered.



### Page 3: Game won/ Game Lost

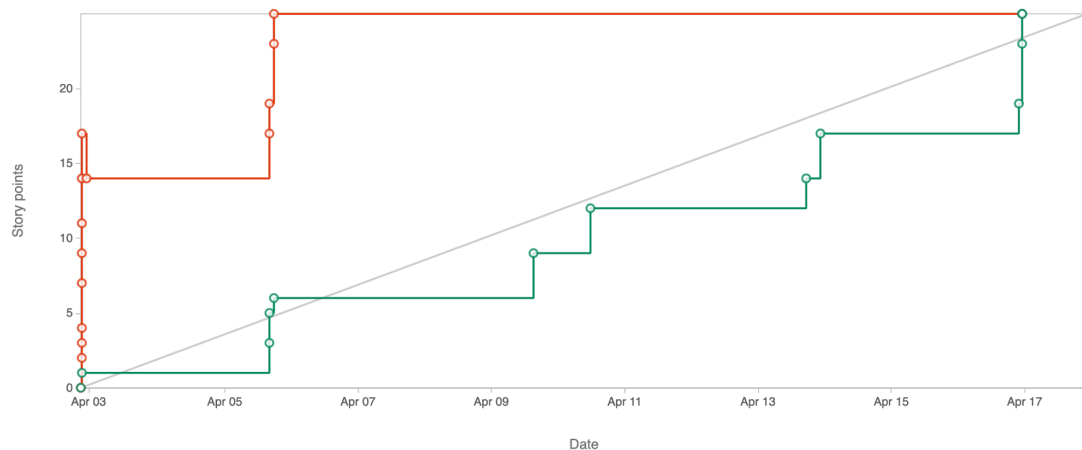
Elements:

- Title → Hoverable feature which changes color upon hover.
- Shape → HTML shape that appear based on number of errors
- Hidden word → Unveils the correct word regardless of success or failure
  - The word is color dependent on the user's guess. Green for correct letters and red for letters not guessed.
- Text → Displays number of errors
- Button (Type: Submit) → Returns user back to home page (Refer to Page 1 wireframe)
- Button (Type: Submit) → Allows user to play the game again (Refer to Page 2 wireframe)
- Text → States if the game was won or lost

## 2.4 Burnup report and Burndown chart

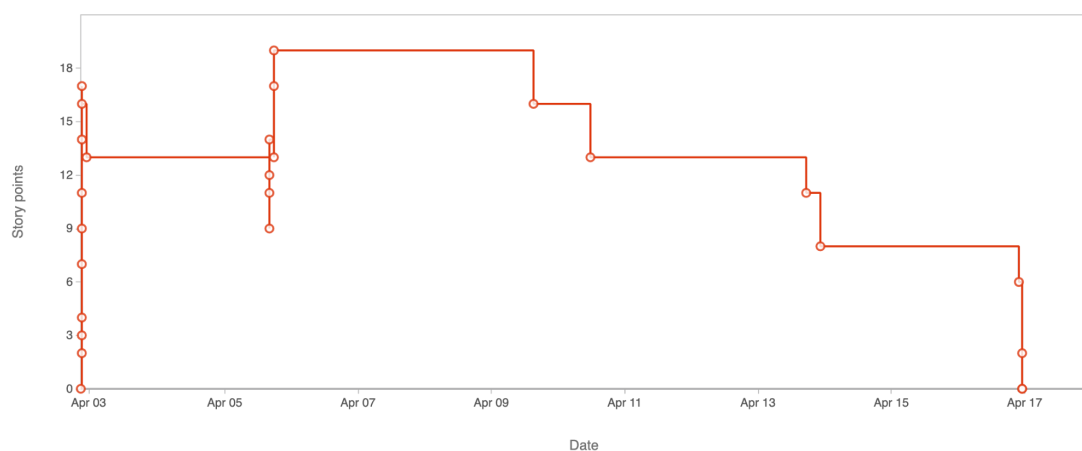
### Burnup Report

The following is the burnup report for the number of tasks created and completed during the project. The red line in the burnup report shows the number of tasks that needed to be completed. At the same time, the green line indicates the number of tasks to be completed. The goal was achieved since the green line and red line meet, showing the sprint period's end. According to the burnup report, the workboard sprint lasted a duration of 14 days throughout April 3rd to April 17th. The burnup report shows the progress per day was consistent. Even Though, the progress was slow in the start, it can be argued that the number of tasks in the beginning were less as it was just the planning phase.



### Burndown Chart

The team rate of production in the start was too slow and forced us to catch up on the work on the last leg of the sprint.



## **2.5 What you want to modify**

A background image should be added to the home page. An error message should be displayed if the user hasn't input their name. The title should be changed.

A new drop-down input "Difficulty Level" should be added for better user experience allowing the user to select their choice of difficulty. Then a random word is selected based on its length. A sound can be added every time we click the play button. The leaderboard should be edited to account for the difficulty.

The name of the user should be stored in such a way that only the first letter is capital.

The errors should only be counted after the user guesses at least one letter of the word.

The user should have a choice to either guess one letter or the whole word (the game ends if the user guesses the whole word wrong).

The webpage should be opened automatically when you run the game.

## 3. Implementation

### 3.1 Code modification Reason

The following modifications are listed and categorised based on the requirement analysis. Each modification has a status of whether it was implemented or not and the reason for which it was implemented. These are lists of modifications made to the original template provided.

#### Front End Modifications:

Modification	Reason	Status
Background Image	To make the game aesthetically pleasing and improve the user interface.	Implemented
Typography	To make the game aesthetically pleasing and improve the user interface.	Implemented
Colour Scheme	Improve user interface and colour coordinate the entire game.	Implemented
Alerts	Informs the user if something is done wrong. This would be useful when a user does not enter their name in the input field and presses play or when a user enters a letter that already exists.	<u>Implemented</u> : Alert for form submission without input entry. <u>Not Implemented</u> : Alert for user inputs a letter that already exists.
Sound Effects	To add user experience to the game every time the user plays.	<u>Implemented</u> : sound effect for each button click  <u>Bugs</u> : Response time delay with the sound when button is clicked which brings up issues with user interface. Click sound effect doesn't



		play when the user presses enter on the keyboard.
		<u>Not Implemented</u> : sound effect for wrong letter entered by user.
Change Hangman layout	To improve user interface and make the game more appealing	Cannot be implemented (Javascript function is not understood by any team members.)
Dropdown button	Allows users to choose difficulty levels: easy, medium, hard.	Implemented
Hoverable features	Improves user experience. Hoverable feature to buttons and title	Implemented
Labels	Improves usability and the user understands what to input in each input field.	Implemented
Leaderboard for Easy, Medium and Hard	Allows users to see their score based on their difficulty level.	Almost Implemented (Back end Code required)
Play button removed	The button is removed so that the user can change the difficulty.	Implemented

### Back End Modifications:

Modification	Reason	Status
Difficulty Level	The user can play an easy, medium or hard game. Allows users to choose which level they want to play.	Implemented
Enter Word or Letter	Allows the user to finish the game off more easily if the correct word is guessed and if the wrong word is entered the game ends, which increases usability.	Implemented
Errors are only counted after the first correct letter is inputed.	The errors the user makes are not counted, which allows the user to get at least one letter before they lose.	Implemented Bugs: While the user is playing the game, the user only sees the errors after he guesses one letter. This contributes to user experience and interface issues.
Open the webpage Automatically	The webpage is opened automatically when the user opens the app.	Implemented
Change the way the name is stored	The name is stored in such a way that the first letter is capitalised	Implemented

## 4. Quality Assurance

### 4.1 User Experience



User experience was an important aspect for our program and we wanted to make it as user friendly and engaging as possible whilst adding a twist to the basic functionalities other than just being able to guess one letter. To do so, we asked an external person who has no experience with the game to provide some feedback regarding the difference between the original game from the template and the modified game. The main objective of this was to see if the new game improves the user experience from the aspect of aesthetics and functionality.

From the first picture provided, we can see that the test subject tried out the original hangman game that was provided to us. Afterwards, she told us that the game seemed too plain, lacked player engagement, and further commented on how the white spaces background made the game seem too basic.

From the second picture provided, the test subject then tried out our modified game and was surprised at how sophisticated the background graphics and user interface looked as some hoverable colour changing features were added to improve the goriness of the “hangman”. Afterwards, she said that the gameplay was improved due to the functionality of being able to guess multiple letters. Although the game was a chance of luck and skills at the same time, being able to guess the whole word excelled the guessing aspect to another level.

## 4.2 Test case for original program

The following are the test cases for the original program conducted, including those test cases where no bugs were detected.

Test cases:	Results:
Inputting letters in name box	The program will register the name and display on the leaderboard.
Inputting numbers in name box	The program will register the name including the number/s and display them on the leaderboard.
Inputting special characters in name box	The program will register the name including the special character/s and display them on the leaderboard.
Clicking play button	Correctly initiates game session
Inputting numbers in word box	The program ignores the inputs
Inputting special characters in word box	The program ignores the inputs
Inputting letters that we've inputted already	The program ignores the inputs

Input more than one character	The program only allows one character for input, and any extra characters will not be shown
Inputting correct letter into the program	The letter is displayed on one of the blanks, no changes made to hangman
Inputting incorrect letter into the program	The letter is displayed near the errores text field. Error count is incremented by one and a body part of the hangman appears.
Checking state if 6 errors are made	The game ends and the user loses, but the user has an option to either play a new game or go back to the home page.
Guessing the correct word	Upon guessing the correct word, the user wins and has the option to play a new game or go back to the home page.

### 4.3 Test case for modified program

The test cases for the modified program are an addition to the test cases to the original program, meaning that each test case from the original program was also checked in the modified program. These testcases have been listed in the table below, which includes those test cases where no bugs were detected.

Test cases:	Results:
Inputting letters in name box	The program will register the name and display on the leaderboard.
Inputting numbers in name box	The program will register the name including the number/s and display them on the leaderboard.
Inputting special characters in name box	The program will register the name including the special character/s and display them on the leaderboard.
Clicking play button	Correctly initiates game session
Inputting numbers in word box	The program ignores the inputs
Inputting special characters in word box	The program ignores the inputs
Inputting letters that we've inputted already	The program ignores the inputs

Input more than one character	The program only allows one character for input, any extra characters will not be shown
Inputting correct letter into the program	The letter is displayed on one of the blanks, no changes made to hangman
Inputting incorrect letter into the program	The letter is displayed near the errores text field. Error count is incremented by one and a body part of the hangman appears.
Checking state if 6 errors are made	The game ends and the user loses and has an option to play a new game or go back to the home page.
Guessing the correct word	Upon guessing the correct word, the user wins and has the option to play a new game or go back to the home page.

## 5. User Manual

### **Description:**

A game where you guess a mystery (hidden) word one letter at a time within a certain limit of tries. Each time the wrong letter is guessed a part of the man appears, when every single body part appears on the man, he is hanged and the game ends.

### **Objective:**

The objective of the game is to guess the hidden word before the number of tries run out (or before the man is hanged). In this case the user has a maximum of six possible errors before the man is hanged and the user loses.

### **How to play:**

1. Begin with entering your name in the input field below *Enter your name:*
2. Choose your desired difficulty by selecting the dropdown menu under *Choose Difficulty Level:* and choose between three modes (*easy, medium, hard*)
3. Click the *play* button when the name and desired difficulty is inputted, the game should then begin.
  - 3.1. Note: If name is not selected an alert will pop up.
  - 3.2. Note: If difficulty level is not selected it will automatically be set to easy.
4. In the text field, start by inputting the letters that could possibly be a part of the word. You can enter letters one at a time, or enter an entire word that is the same length as the hidden word.
  - 4.1. Note: The game is lost if you enter the wrong word, or if you enter 6 wrong letters.
  - 4.2. Note: If the word that is imputed is not of the same length as the hidden word, the word will be ignored. (Refer to Rule 3.1)
5. If the word is guessed correctly, the game is won.
6. Once the game is finished (regardless of a win or loss) you can choose to *play again* or return to *home* by clicking the respective buttons.
7. If the *play again* button is selected, the game begins again. (Refer to how to play step 4)
8. The leaderboard may display the player's name if the player's game points are among the top 10. (Refer to Rule 5 to see how game points are calculated)

### **Rules:**

1. The game does not start if the player's name is not entered.
2. The game only begins to count errors after the first letter is guessed correctly, errors will be displayed but will not be counted towards six possible errors.



3. In the input box:
  - 3.1. One letter can be inputted at a time or the entire word of the same length as the hidden word (the length of the hidden word can be noted by the number of underscores ' \_ ').
4. You lose the game when:
  - 4.1. You run out of tries while guessing the letters in the word
  - 4.2. You enter an incorrect guess of the word with the same length as the hidden word.
5. Game points are calculated based on the length of the word and the number of errors. The leaderboard displays the player name, the word, the errors made and the game points for players ranked in the top 10.

**Modes:**

The game includes 3 difficulty modes (easy, medium, hard):

- Easy allows for 6 tries for a 4 to 6 letter word.
- Medium allows for 6 tries for a word between 6 to 9 letters.
- Hard allows for 6 tries for words longer than 9 letters.