Washington State University
Vancouver

Software Requirements Specification

for

HangMan

Version 1.0

Prepared by

Group Name: CupaRum

David Barko	<student #>	David.barko@wsu.edu
Daniel Yarmolenko	<student #>	Daniel.yarmolenko@wsu.edu

Date: <place the date of submission here>

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft Type and	Full Name	Information about the revision. This table does not need to be filled in whenever a document is	00/00/00

Version	Primary Author(s)	Description of Version	Date Completed
Number		touched, only when the version is being upgraded.	

<In this template you will find text bounded by the “<>” symbols. This text appears in italics and is intended to provide explanations and guide you through the document. There are two types of comments in this document. The comments that are in black are intended specifically for the course. The comments that are in blue are more general and apply to any SRS. Please make sure to delete all of the comments before submitting the document.>

1 Introduction

1.1 Document Purpose

The purpose of this document is to provide a clear and concise description of Hangman to minimize conflicts in the implementation process of this project. In short, HangMan is a word game aimed to provide users with a fun way to challenge themselves while exercising their problem-solving skills. Also, users can create their own challenges to share with others. This document covers Hangman's functionality and design constraints, specific requirements such as user interfaces and software interfaces, and non-functional requirements. Moreover, the SRS document provides context and use case diagrams to illustrate the intended use of Hangman while also explaining the expected user interactions.

1.2 Product Scope

Hangman is an interactive web application word game that provides users with challenges in a fun and educational way. The objective of Hangman is for the user to guess the word or phrase that the game is hinting at without getting more than 6 letters incorrect, which then completes the noose. For the hint, the program will provide a one sentence description of the word or phrase. Also, an alphabet of letters will be provided where the player can click on a letter that they think is in the phrase. The main benefits of this game is that it will help users expand their vocabulary and/or spelling, and exersice their problem solving skills but at the same time allow users to have fun. On top of that, this software allows users to share their experience with friends and family by challenging them in the word game. Users can challenge others by uploading their own hints and answers then have their peers try to get all the answers without loosing.

1.3 Intended Audience and Document Overview

This Software Requirements Specification document is intended for the client and the professor. It is written for the readers to understand all the specific requirements of the product and the final developed software. Providing all the information about the product in a very specific manner, touching on all the intended topics at hand. This document is to be used to help the client understand the purpose and the complete overall design of the final software product Hangman. The professor may use this document to get a overall view of the specific requirements to this software. The Specific requirements of this software can be found throughout the rest of this document.

1.4 Definitions, Acronyms and Abbreviations

Challenge – A Challenge includes a hint and the answer to the hint. The hint is a sentence that gives the user a clue as to what the answer is where the answer is a word or a phrase.

Hangman – The name of hangman refers to a character that gets hanged by a noose if the player chooses a certain amount of incorrect letters. Each incorrect letter draws a part of the hanging scene, once complete the hang man dies and the user looses.

Hint – A hint is a short sentence or phrase that gives a user a clue of what the word is.

1.5 Document Conventions

This document follows the IEEE citation and formatting requirements, and bold text is used to emphasize sub-headings of each section.

1.6 References and Acknowledgments

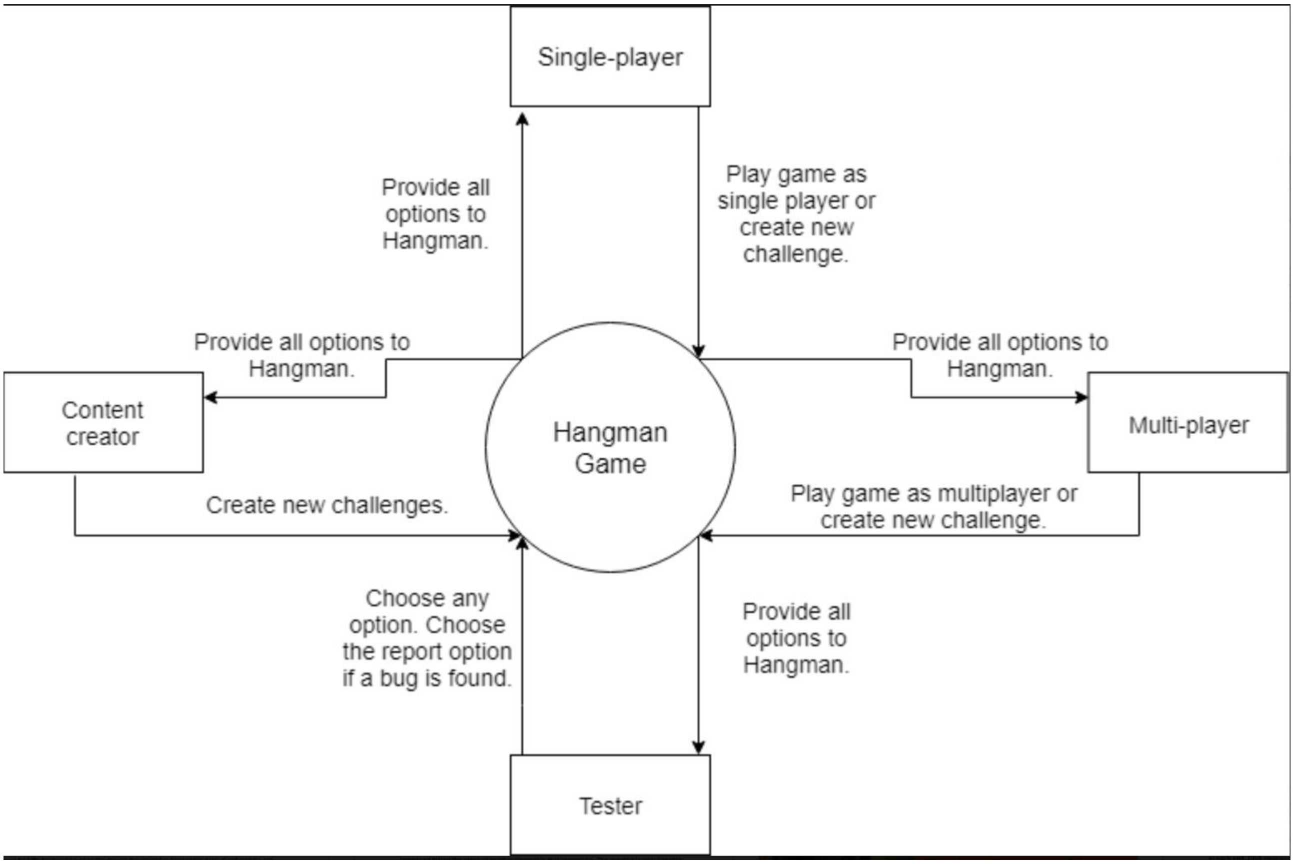
We used Draw.io to help us create our diagrams.

Draw.io. (2019). Flowchart Maker & Online Diagram Software*. [online] Available at: https://www.draw.io/ [Accessed 26 Oct. 2019].*

2 Overall Description

2.1 Product Perspective

Hangman is a new, self-contained product aimed at providing a fun way to exercise the user's problem-solving skills with words. The game can be controlled by the user by allowing them to create their own clues for certain phrases by typing them into the corresponding text boxes provided. Then the user can share their new challenge to others. In a teacher's perspective, they can add their own challenges that are relevant to a concept being taught and then share it with other students. These challenges can be uploaded onto the web application and the device the game is played on is given to others so that they can attempt to solve it without losing. Otherwise, students or other users can attempt challenges that are premade already or challenges made by other users which are chosen at random by the game.


Single-player
Provide all options to Hangman.
Play game as single player or create new challenge.
Provide all options to Hangman.
Content creator
Hangman Game
Provide all options to Hangman.
Multi-player
Create new challenges.
Play game as multiplayer or create new challenge.
Choose any option. Choose the report option if a bug is found.
Provide all options to Hangman.
Tester

2.2 Product Functionality

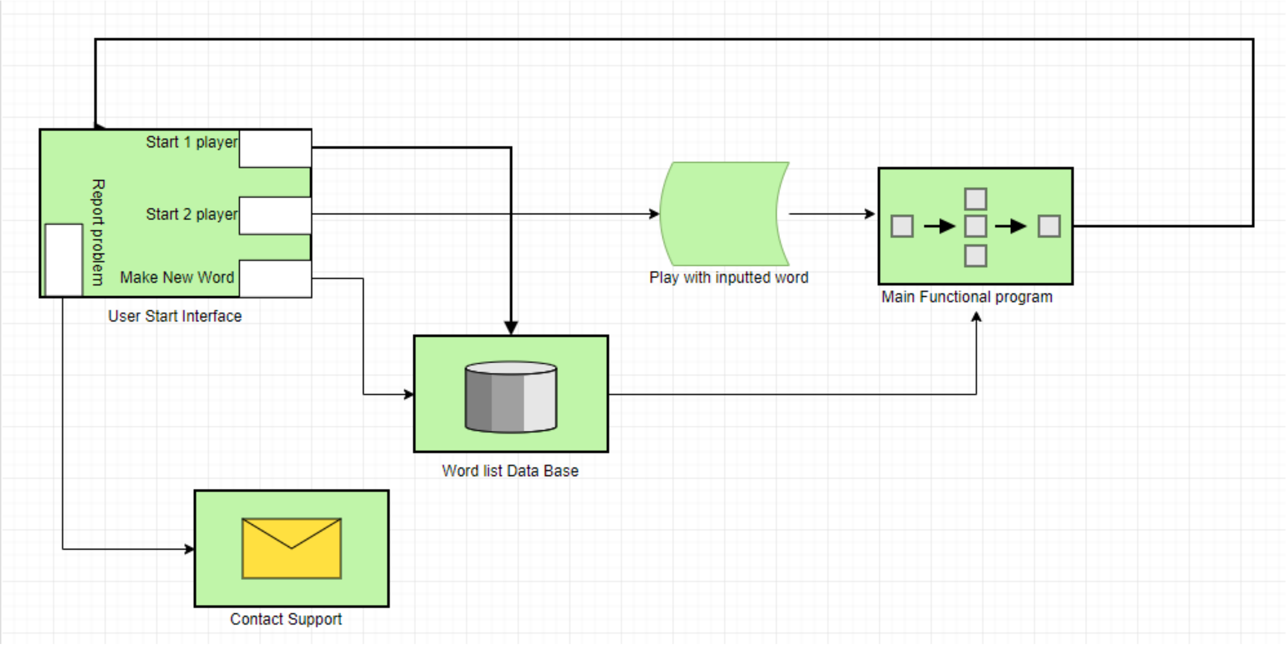
This part of the document provides a high-level overview of the products functions in the form of a list.

- Welcome page includes a small description of what hangman is and a tutorial on how to play.
- Below the description there are 3 options:
  - Play single player
  - Play multiplayer
  - Upload new challenge
  - Report error or abuse
- If multiplayer is chosen a user creates a new challenge and gives the device to another player so they can attempt to solve the challenge.
- Game start when single player or multiplayer is chosen.
  - A random hint of the phrase or word is given at the top.
  - Alphabet of all 26 let
  - A new graphic is added of Hangman if users choose the incorrect letters.
  - If user guesses all the letters in the phrase before getting 6 incorrect, they win the challenge.
  - If the user gets 6 letters incorrect they loose the challenge and the correct answer is shown to them.
- If user is sharing a new challenge, they will write the word and the hint into dedicated text boxes. The challenge is then added permanently to the database.
- User has the option to play a different challenge or upload a custom challenge, or logout and exit.

2.3 Users and Characteristics

All users have the same accessibility to the features of Hangman. However, this software can be utilized for many things. For example, teachers can use the game as way to help students memorize or learn about certain concepts by uploading customized challenges. Students can use this software to study by creating their own challenges for others in a study group which can help them with memorizing concepts. Younger audiences that are under 13, with parental supervision, can also use this game as way to spend time learning and having fun. Furthermore, everyone else can play to pass time, for fun or to for self-improvement and not just for educational purposes. For instance, Hangman is great for social party games where the participants try to guess who in the party is the hint is referring to.

2.4 Operating Environment



This software will be able to operate on most basic web browsers and won't look at any specifications of outside hardwire. This software's main code will be in Javascript. However, it will be designed with CSS and html. The rest of the software development will be with the CSS and html which is considered the front end of the web application. JavaScript will also be used to manage some of the inputs and outputs and basic data handling that the main game will give out.

2.5 Design and Implementation Constraints

The issues this software might have is with most outdated systems that won't run any of the newer web applications. Also, the game requires an internet connection so offline use is not possible. Development wise, the software runs using JavaScript, html 5, and CSS and produced using IntelliJ IDE. A big challenge is producing a web application that runs smooth because the hangman game requires a lot of code and data manipulation in JavaScript. Especially with older and weaker hardware, the game will have a harder time being fluid in execution because of ram limitations or low network speeds.

For the front-end implementation of the web application, jQuery library for JavaScript and the bootstrap framework will be used to simplify the code for front end design. Another constraint is the device screen size of the user's device, it can affect the game execution if it is too small, e.g a mobile phone or small tablet. Therefore, the game will not be able to run on mobile since it is designed to work on desktop computers, however future versions will add mobile compatibility.

2.6 User Documentation

The user manual will be simple and straight forward. The software will deliver all the important instructions on how to play and use the software. The software will also come with a report a problem link which will allow the user to contact the developers and report an prolem or ask any questions realting to the software. Other than the instructions and a contact forum page to report problems, the software will just come as a basic package.

2.7 Assumptions and Dependencies

The web application is design assumes that the player is using a desktop computer when they play the game. The game will be affected in unpredictable ways when the screen is too small and touch controls are used instead of a mouse. This is because during the implementation of the game, touch controls and mobile users will not be considered until future versions. Furthermore, it is assumed that the user uses an up to date web browser that allows the use of html 5 and bootstrap 4. The use of older web browsers will not be considered in the implementation of this game.

Another assumptions is if a user uploads a challenge, it provides answers which are relevant to the hint. If the challenges are too difficult or purposely made to be impossible to solve, the users experience of the game may become unpleasant. Also, it is assumed that the uploaded hints are not offensive to the general public. The purpose of the game is to have fun and learn, however if the challenges make no sense, or are just there to send an inappropriate message, it will change the game goals.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The graphics of the software will be simple and straight forward and easy to navigate across. The main screen will print the current status of the hangman and there will be a keypad where you will be able to enter the letters of the English alphabet to guess the word. There will be a start game button and a button that says resign which after the user will be prompted to make sure that they want to quit. The logic behind the keypad will be to insert the chosen letter into the JavaScript to check if that letter is in the word or phrase. After which the letter will promptly disappear from the keypad to reassure the user that any chosen letters will not be repeated. The graphical drawings of the hangman will be basic and simple, allowing the user to also see their current health status and current wrong guesses left before they lose. The software will also all the user to input the whole word if they think that they have the whole word right. However, by doing so this risks the rest of the turn resulting in an automatic lose if they guess the full word incorrectly. Also there will be a button that will allow the user to add their own word, the software will check the database of words amnd phrases to make sure that the word doesn't have duplicates.

3.1.2 Hardware Interfaces

The keypad will have a characteristic where when pressed, it will insert that letter into the JavaScript code and run through the linked list of data and checking all the repetitions of that letter in the word or phrase. The keypad button will then disappear after being clicked. In the beginning the user will press the button that starts the game and in game there will be a button to resign and quit, followed by a short JavaScript prompt that will ask for assurance on the users choice as all data will be deleted if the game is quit mid progress. The graphics portion of this software will just display the basic status of the game and according to the JavaScript code, the current position of the hangman.

3.1.3 Software Interfaces

This software will be able to run on linux, windows and mac. Any updated web browser should be able to run this software. The data base will utilize SQL, and Javascript will be the main software language held within HTML and designed with CSS.

3.1.4 Communications Interfaces

The software will come with a report a problem forum that allows the user to contact us through a email messeging system embedded in the web application. For security purposes to avoid spam and other unwanted bot nets the software will come with an authentication check that requires the user to complete a simple or complex captia to prove their realness and idenity before they will be able to contact the developers.

3.2 Functional Requirements

3.2.1 Landing Page

The landing page includes a small paragraph that describes what Hangman is. Below the description are three options in the form of buttons, one is to play a single player game, the second one is to play multiplayer, and the third is to upload a new word and hint into the game.

- Option: single player:

  If the option for playing the game as single player is clicked, the user will be brought to the game with a random challenge.

- Option: multi-player:

  If the multiplayer option is clicked the user is brought to the game with text boxes provided for the word/phrase and for the hint. When the challenge is added the game begins and the user can give the device to another so they can attempt to figure out the word.

- Option: upload new challenge:

  If the player chooses to upload new challenge, they are given dedicated text boxes to add the word/phrase and to add a hint. The once the user has clicked on done, the challenge is added permanently into the database of the game.

- Option: Report

  The report button will allow a user to send a message to the developers if they experienced a bug or if the content of the challenges is offensive.

3.2.2 Game Play

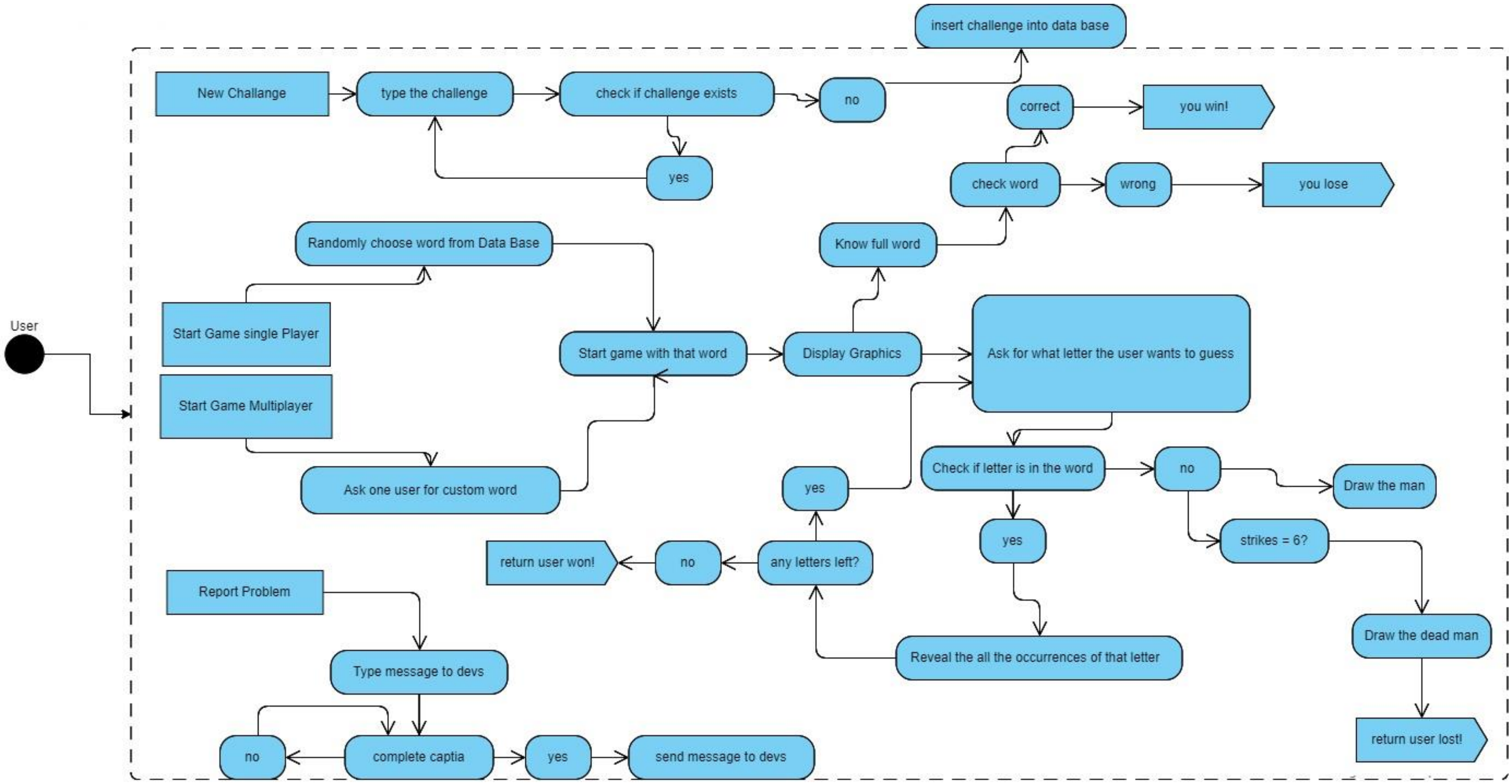
The game begins after the single player button is clicked, or after the multiplayer button is clicked and player 1 adds a new challenge for player 2 to attempt.
Gameplay:

- A random hint is given at the top of the game screen.
- The base for the hang man graphics is shown in the game screen and below it there are blank underlines for each letter of the word/phrase that needs to be guessed.
- An alphabet of all 26 letters is provided to the user. To choose a letter, the player needs to click on that letter.
- If the letter is correct, the letter will show up on its corresponding underline space in the word/phrase. If the letter is incorrect, a new graphic of the Hangman is drawn.
- In the case that the user guesses 6 incorrect letters, the Hangman graphic completes and the users loose. The correct answer is then given to them.
- In the case that the user guesses the complete word with less than 6 incorrect letters, they win the challenge and are congratulated via a message in text.

At any point during the game the player may go back to the landing page menu, however once they leave that challenge the progress of the game will be lost. Also, the challenge might not be the same one when they play again as the challenges are given at random.

3.3 Behaviour Requirements

3.3.1 Use Case View



Use Case Diagram

4 Other Requirements

4.1 Performance Requirements

The preformance of the software should be quick and simple to save on loading time. The user interface has to be simple to navigate and the calculations between the data base and the code that manipulates it cannot take too long to execute. When a letter is guessed the software has to compute the next outcome quickly and efficiently. If the user wants to enter a word or phrase into the data base the full computation shouldn’t take long than 2-5 minutes depending on how large the data base is. Each function should be executed when it is required with no delay.

4.2 Safety and Security Requirements

1. *The software must only except safe input, to avoid damage, loss of data and infiltration of the data base.*
2. *Software will assume that the user wants to input injections*
3. *Safe check any user input before executing it.*
4. *Don’t allow any user input manipulate the data base.*
5. *A captia will be used to verify users who want to contact the developers.*

4.3 Software Quality Attributes

The attributes of this software is to give users a limited amount of input to avoid any injections of malicious code. The software will always allow the user to report any noticed bugs. The software will have a unique atribute to allow a growing data base which makes the software always unique and fan made.This software will be very easily usable and easily maintainable.

Appendix A – Data Dictionary

<Data dictionary is used to track all the different variables, states and functional requirements that you described in your document. Make sure to include the complete list of all constants, state variables (and their possible states), inputs and outputs in a table. In the table, include the description of these items as well as all related operations and requirements.>

Appendix B - Group Log

<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist the Teaching Assistant to determine the effort put forth to produce this document>