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

<TO DO: Please provide a brief introduction to your project and a brief overview of what the reader will find in this section.>

1.1 Document Purpose

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.
TO DO: Write 1-2 paragraphs describing the purpose of this document as explained above.>

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

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals.
TO DO: 1-2 paragraphs describing the scope of the product. Make sure to describe the benefits associated with the product.>

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

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers (In your case it would probably be the "client" and the professor). Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>
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

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.
TO DO: Please provide a list of all abbreviations and acronyms used in this document sorted in alphabetical order.>

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

<In general this document follows the IEEE formatting requirements. Use Arial font size 11, or 12 throughout the document for text. Use italics for comments. Document text should be single spaced and maintain the 1" margins found in this template. For Section and Subsection titles please follow the template.

TO DO: Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. Sometimes, it is useful to divide this section to several sections, e.g., Formatting Conventions, Naming Conventions, etc.>

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

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document.

TO DO: Use the standard IEEE citation guide (attached) for this section.>

Acknowledgment goes to professor grey, scott koss, nick macias and ronald reagan. Special thanks to the chemicals in the water.

2 Overall Description

2.1 Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. In this part, make sure ***to include a simple diagram that shows the major components of the overall system, subsystem interconnections, and external interface. In this section it is crucial that you will be creative and provide as much information as possible****.*
TO DO: Provide at least one paragraph describing product perspective. Provide a general diagram that will illustrate how your product interacts with the environment and in what context it is being used, i.e., context diagram.>

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 uploading a text document that follows a specified template for this game. 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.

2.2 Product Functionality

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, will be effective.
TO DO:
1. Provide a bulleted list of all the major functions of the system
2. ***(Optional)*** *Provide a Data Flow Diagram of the system to show how these functions relate to each other. This is useful when there is a clear sequence for the functions being performed.>*

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
- 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

<Identify the various users that you anticipate will use this product. Users may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience.
TO DO:
1. Describe the pertinent characteristics of each user. Certain requirements may pertain only to certain users.
3. Distinguish the most important users for this product from those who are less important to satisfy.>

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

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist. ***In this part, make sure to include a simple diagram that shows the major components of the overall system, subsystem interconnections, and external interface***
TO DO: As stated above, in at least one paragraph, describe the environment your system will have to operate in. Make sure to include the minimum platform requirements for your system. >

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

<Describe any items or issues that will limit the options available to the developers. These might include: hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).
TO DO: In this section you need to consider all of the information you gathered so far, analyze it and correctly identify relevant 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

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.
TO DO: You will not actually develop any user-manuals, but you need to describe what kind of manuals and what kind of help is needed for the software you will be developing. One paragraph should be sufficient for this section.>

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

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project.
TO DO: Provide a short list of some major assumptions that might significantly affect your design. For example, you can assume that your client will have 1, 2 or at most 50 Automated Banking Machines. Every number has a significant effect on the design of your system. >

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

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., Cancel) that will appear on every screen, error message display standards, and so on. Define the software components for which a user interface is needed.
TO DO: The least you can do for this section is to describe in words the different User Interfaces and the different screens that will be available to the user. Optional: You may also provide an initial Graphical User Interface design (does not have to be final).>

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

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware. You are not required to specify what protocols you will be using to communicate with the hardware, but it will be usually included in this part as well.
TO DO: Please provide a short description of the different hardware interfaces. If you will be using some special libraries to communicate with your software mention them here. In case you have more than one hardware interface divide this section into subsections.>

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

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems (Windows? Linux? Etc…), tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.

TO DO: The previous part illustrates some of the information you would usually include in this part of the SRS document. To make things simpler, you are only required to describe the specific interface with the operating system.>

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

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.

TO DO: Do not go into too much detail, but provide 1-2 paragraphs were you will outline the major communication standards. For example, if you decide to use encryption there is no need to specify the exact encryption standards, but rather, specify the fact that the data will be encrypted and name what standards you consider using. >

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

< Functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks or functions the system is required to perform. This section is the direct continuation of section 2.2 where you have specified the general functional requirements. Here, you should list in detail the different product functions with specific explanations regarding every function.

TO DO: Break the functional requirements to several functional areas and divide this section into subsections accordingly. Provide a detailed list of all product operations related to these functional areas.

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.

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

<A use case defines a goal-oriented set of interactions between external actors and the system under consideration.

TO DO: Provide a use case diagram which shows the entire system and all possible actors. Do not include detailed use case descriptions (these will be needed when you will be working on the Test Plan), but make sure to include a short description of what every use-case is, who are the actors in your diagram.>

4 Other Non-functional Requirements

4.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.
TODO: Provide relevant performance requirements based on the information you collected from the client. For example you can say "1. Any transaction will not take more than 10 seconds, etc…>

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

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied. Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements.
TODO:

- *Provide relevant safety requirements based on your interview with the client or, on your expectation for the product.*
- *Describe briefly what level of security is expected from this product by your client and provide a bulleted (or numbered) list of the major 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

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.

TODO: Use subsections (e.g., 4.3.1 Reliability, 4.3.2 Portability, etc…) provide requirements related to the different software quality attributes. Base the information you include in these subsections on the material you have learned in the class. Make sure, that you do not just write "This software shall be maintainable…" Indicate how you plan to achieve it, etc.>

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.

5 Other Requirements

*<This section is **Optional**. Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>*

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>