

Software Requirements Specification

for

Maze Web Application

**Version 1.0**

**Prepared by**

**Group Name: *Team 19***

|  |  |  |
| --- | --- | --- |
| **Patric Collins** | **11617373** | **patric.collins@wsu.edu** |
| **Matthew Norvell** | **11690711** | **matthew.norvell@wsu.edu** |
| **Andrew Keyes** | **11595911** | **andrew.keyes@wsu.edu** |
| **Daniyal Abbas** | **11714220** | **daniyal.abbas@wsu.edu** |

|  |  |
| --- | --- |
|  |  |
| **Date:** | **2020-11-06** |

Content

**Revisions 3**

**1** **Introduction 4**

1.1 Document Purpose 4

1.2 Product Scope 4

1.3 Intended Audience and Document Overview 4

1.4 Definitions, Acronyms and Abbreviations 5

1.5 Document Conventions 5

**2** **Overall Description 6**

2.1 Product Perspective 6

2.2 Product Functionality 6

2.3 Users and Characteristics 6

2.4 Operating Environment 6

2.5 Design and Implementation Constraints 6

2.6 User Documentation 6

2.7 Assumptions and Dependencies 7

**3** **Specific Requirements 7**

3.1 External Interface Requirements 7

3.2 Functional Requirements 8

3.3 Behavior Requirements 9

**4** **Other Non-functional Requirements 9**

4.1 Performance Requirements 9

4.2 Safety and Security Requirements 10

4.3 Software Quality Attributes 10

**Appendix A – Data Dictionary 11**

**Appendix B - Group Log 12**

**Revisions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Primary Author(s)** | **Description of Version** | **Date Completed** |
| Final  Draft 1 | Patric Collins  Matthew Norvell  Andrew Keyes  Daniyal Abbas | Initial revision of the final draft. | 11/06/20 |
| Final Draft 2 | Andrew Keyes  Matthew Norvell | Updated document to be better in line with the final product. | 12/16/20 |

# Introduction

## This project is the final project for CS 320. Our group of four people has created a website that meets the guidelines for the project by handling and manipulating data.

## Document Purpose

This SRS document will cover the entirety of the maze program it outlines. The purpose in including all of this is to give a full view of what the program will look and act like when complete. This program is currently in development, but once complete will be Release No. 1. References were not needed as we did not consult any outside sources.

## Product Scope

The product is a website that allows users to play randomly generated mazes. These mazes have different difficulties that can be chosen by the user. Users will then have their scores recorded and their maze saved for future use. The recorded scores are displayed on a leaderboard for users to view. The score will include a maze generation seed that can be used to generate the same maze as another user, encouraging competitive play.

## Intended Audience and Document Overview

This document is meant to be an overview for any future admins of the website and Professor Hedden. This SRS contains a description of the website, specific technical requirements, and other non-functional requirements. It is intended to be read in page order by future admins to guide the reader gradually into more specific detail over time. Professor Hedden may need to read “Specific Requirements” (Section 3) first to better understand how our website works at a glance.

## Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| HTML | Hypertext Markup Language, used to format web pages |
| JavaScript | Programming language used to interact with webpages |
| Node.js | A runtime for JavaScript |
| Score | The time taken by a user to complete a maze |
| Seed | A number used by the website to generate a maze. |
| Sprite | The image that represents the player navigating the maze. |
| Username | A unique identifier for a user. |
| Whoosh | A Python search engine library. |

## Document Conventions

In this document, these conventions were followed:

* Arial font size 11 for content
* Arial bolded font size 18 for section headers
* Arial bolded font size 14 for subsection titles
* Arial bolded font size 12 for sub-subsection titles
* Single spaced content
* 1” margins on all sides
* Justified alignment for content
* Left alignment for section and subsection titles

(1.6 is missing because we did not reference any other documents in the SRS document.)

# Overall Description

## Product Perspective

This is a self-contained product and will not be a replacement or extension of another system or product. The product will require JavaScript in order to be developed for a website. Interaction with the product will involve multiple users playing the maze game and will be able to show how well they performed or replay the mazes that they attempted.

## Product Functionality

* User can either play the maze game or search another username
* If the user chooses to play the game, then they will play it and how well they performed is recorded
* If the user chooses to search another username, then the product will return that user’s performance

## Users and Characteristics

The users that are expected to visit the website can be anyone due to how simple the maze game will be. Anyone on the website should be able to understand that the product is a maze game. A key characteristic of users is the frequency of use due to recording scores and previous mazes. Thus, the essential users are those that return to the website over those that only visit once.

## Operating Environment

The software is being developed for a website and can be able to run anything with internet access. The focus will be on desktop users accessing the website. A required component for this product is the use of JavaScript to develop it.

## Design and Implementation Constraints

An issue for this project is how it will record previous mazes and how well users performed on them which means a database is needed to store data. The main language for the development of this product is JavaScript due to the website needing to display and let the user interact with the maze.

## User Documentation

The instructions for how to use the product will be provided as the user progresses through the website. The user will be told how to set up the game such as the difficulty of the maze and how to navigate through it. At the leaderboard the website will let the user know how to look up other users for their scores and previous mazes and will be given the option to replay them.

## Assumptions and Dependencies

It is assumed that the user will either play a few different mazes about 1 to 3 times or more than 10 times. This may need to be considered by deciding if the previous mazes for the users should all be recorded or only about 10 should be kept on record. For users that only play one maze game and never come back to play another, it needs to be decided how long their scores should be saved.

# Specific Requirements

## External Interface Requirements

### User Interfaces

#### Landing Page

This page is what the user will first see upon startup of the application. There will be two options available to the user on this page. The first option allows the user to transition to the maze difficulty selection page and the second allows the user to transition to the username search page.

#### Maze Difficulty Selection

This page is where the user will be able to select what difficulty of maze to generate. This page will have multiple buttons that allow the user to select different difficulties. Once the difficulty is selected the user can then move onto the maze running page. There will also be an option to return to the landing page.

#### Maze

This page is where the user will play the maze generated for them based on the difficulty selected on the previous page. The interface will contain the maze to be played through and a timer to show the user how long they have taken to complete the maze. Once the maze is completed the user will be prompted to move to the post maze completion interface. The user will also have the option to return to the maze difficulty selection screen.

#### Maze completion screen

This page is shown to the user after the maze is completed. This page allows the user to insert a username that will be saved along with the maze they completed and how long it took to complete that maze. The user also has the option to replay the maze, returning to the maze screen, or to return to the landing page.

#### Search username page

This page allows the user to insert a username into a search bar. The page will then display a list of all mazes that the username is associated with. Each entry will display the difficulty, time to finish, and a button that will allow the user to play that maze, transitioning the user to the maze page. The user also will have the option to return to the landing page.

### Hardware Interfaces

* Display monitor: It is recommended that the user has a modern, high resolution monitor.
* Input devices: Input devices required include a mouse and keyboard. The mouse will be used to interact with the various features of the app pages. The keyboard will be mainly used to play the maze game and used to input usernames and search for usernames.

### Software Interfaces

* The application requires the user run the application on one of the following browsers:
  + Chrome
  + Internet Explorer
  + Microsoft Edge
  + Firefox
  + Safari
  + Opera

### Communications Interfaces

* The application will be communicating with the user through an open web browser.

## Functional Requirements

### Front end Requirements

The application will allow the user to move between five different pages. These pages include a landing page, username search page, maze difficulty selection page, maze page, and post maze completion page.

### Maze Requirements

* The application will have three different maze difficulties based on size: small, medium, and large.
* The application shall generate mazes based on difficulty selected by the user. This maze will be created in a grid filled with two different types of tiles. These tiles will be denoted as walls and paths.
* The user shall control a sprite as they move through the maze.
* This sprite will have unit collision with wall tiles and will be unable to move through them.
* The user will only be able to move along path tiles. Two of these path tiles will be denoted as the start and finish tiles.
* The user’s sprite is placed on the start tile upon generation.
* Once the user leaves the start tile, a timer will start counting.
* This timer will continue until the user’s sprite has reached the end tile.
* Upon generation the maze shall have at least one set of continuous path tiles from the start to the end tile.
* The seed that was used for maze generation will be saved temporarily for potential permanent storage.

### Username Storage Requirements

* Upon completion of a maze, the user shall have the option to save a username tied to the completion time of a specific maze.
* The seed used to generate the maze will also be stored for potential future regeneration.
* The user will be able to search for usernames. The user will then be shown all mazes that were saved with that username.
* The user will then be able to generate and play any maze displayed.

## Behavior Requirements

### Use Case ViewDiagram Description automatically generated

# Other Non-functional Requirements

## Performance Requirements

The maze web application provides the option for multiplayer and single player. Therefore our web app performance depends on the response rate, latency and data storage.

1. The web application must be interactive and the delays involved must be short
2. Transition between interfaces must be smooth
3. In every response from the system there shall be no immediate delays
4. Saving the settings and session must take less than 2 secs
5. Maze shall be generated under 5 sec
6. Saving any information must take less than 10 sec
7. Must include smooth animations

## Safety and Security Requirements

As we are developing a web application, attacks like DDoS attacks and SQL injections pose a threat to our web application. These attacks may cause our web application to crash or even become completely inaccessible. So the web application must be immunized against these kinds of attacks.

## Software Quality Attributes

The software quality depends on various attributes. The software quality attributes for maze web application are as below:

## 4.3.1 Availability

The application must save all data inputted by the user. The application will also allow the user to reset the data storage, allowing for a clean slate.

## 4.3.2 Reliability

The web application must be reliable in case of low speed internet connection. And it must maintain all the standards described in performance requirements.

## 4.3.3 Usability

The web application must be easy to use. There shall be hierarchical structure of options. The option icons must be self-explanatory. The text and images shall be easy to read and visually appealing.

## 4.3.5 Portability

The web application must behave in a similar manner between web browsers. The response, loading saving time must not be affected by the browser.

## 4.3.6 Correctness

Web application features like leaderboard positions, searching a username and user scores must be up to date and shall produce correct results. The same input must not give two different outputs.

**Appendix A – Data Dictionary**

|  |  |  |
| --- | --- | --- |
| *Name* | *Type* | *Description* |
| Difficulty | String | The difficulty of the generated maze. Difficulty is always paired with a seed. |
| Seed | Number | Used by the website to generate a specific maze. A seed can be associated with many usernames and times. |
| Time | Integer | The time, in seconds, a user takes to complete a maze. Time is always paired with a seed and username. |
| Username | String | Unique identifier for a user. A username can be associated with many seeds and times. |

Typical data storage is in the format: Username **→** Seed, Time, Difficulty

**Appendix B - Group Log**

|  |  |
| --- | --- |
| Date & Time | Subject |
| *2020-10-05 (3:00 - 3:45)* | *Brainstorming and deliberation of the project idea. Many ideas were proposed, but a vote favored the maze idea. We fleshed out aspects of the maze and all functionality it had.* |
| *2020-10-19 (3:30 - 4:15)* | *Delegation of tasks for the SRS document. Each member was given a section to complete: Section 1 - Matthew, 2 - Patric, 3 - Andrew, 4 - Daniyal. The Github repo was created for us to commit to.* |
| *2020-10-26 (3:00 - 3:15)* | *A quick check-in between all members to determine progress and any challenges. Section 2 was completed and work on the other sections had begun.* |
| *2020-11-02 (3:00 - 4:00)* | *We discussed what we had each written and disputed any differences of opinion. We still had some things to complete, which were added by the due date.* |