

CS 320 Course Project Final Report

for

**Maze Web Application**

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: | **2020-12-16** |
|  |  |

Content

Contents ii

1 Introduction 1

1.1 Project Overview 1

1.2 Definitions, Acronyms and Abbreviations 1

1.3 References and Acknowledgments 1

2 Design 2

2.1 System Modeling 2

2.2 Interface Design 2

3 Implementation 3

3.1 Development Environment 3

3.2 Task Distribution 3

3.3 Challenges 3

4 Testing 4

4.1 Testing Plan 4

4.2 Tests for Functional Requirements 4

4.3 Tests for Non-functional Requirements 4

4.4 Hardware and Software Requirements 4

5 Analysis 5

6 Conclusion 6

Appendix A - Group Log 7

# 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.

## Project Overview

< A brief description of the project.

TO DO: Write 1-2 paragraphs describing the project. >

Our project is centered around a web application that allows a user to play through randomly generated mazes. These mazes are then accessible by others through a global leaderboard. Section 2 outlines use cases for the website. Section 3 outlines the class structure of the website. Section 4 outlines both state and sequence diagrams for aspects of the website. Appendix A is a record of all meeting information for the project group.

## Definitions, Acronyms and Abbreviations

<Define all the terms necessary to properly interpret the report, including acronyms and abbreviations.

TO DO: Please provide a list of all abbreviations and acronyms used in this document sorted in alphabetical order.>

|  |  |
| --- | --- |
| **Term** | **Definition** |
| DB | Database |
| Score | The time taken by a user to complete a maze |
| Seed | A number used by the website to generate a maze. |
| Username | A unique identifier for a user. |

## References and Acknowledgments

We did not need to cite any sources for our project.

# Design

## System Modeling

< Update your UML diagrams in milestone 2, to reflect the real implementation of this software.

TO DO: Provide an updated version of the UML diagrams, including use case diagrams, sequence (or state) diagrams, activities diagrams, and class diagrams. If you don’t have an updated version, just mention: “our implementation strictly follows the design document (milestone 2)”. >

## Interface Design

<Provide several screenshots to illustrate your interface design.

TO DO:

For each subsystem, pick one or two representative screenshots and paste here.>

# Implementation

## Development Environment

Daniyal: Visual Code for coding, Firefox for website testing

Andrew: IntelliJ IDEA for coding, Google Chrome for website testing

Matthew: Atom for coding, Firefox for website testing

Patric: N/A

Github for version control. HTML, CSS, and JavaScript for project functionality.

## Task Distribution

Daniyal: Timer implementation and difficulty select page

Andrew: Username database storage and search, main page and formatting

Matthew: Maze generation and display

Patric: N/A

## Challenges

We had two major implementation issues. First was the setup of a backend server for a database. We decided to use local storage in the browser for data instead of a whole backend server. Second, one of our teammates, Patric, did not participate in the later stages of the project after contacting him and the professor. There was an additional workload on the remaining members, although we did finish without much pressure.

# Testing

## <*This section is a summary of your testing report>*

## Testing Plan

<Describe your testing plan for the project.

TODO: Give a list of items or functions you want to test, and also a schedule for performing the testing.

Functional Requirements>z

|  |  |  |
| --- | --- | --- |
| Maze generator | Maze is generated | 2020-12-07 |
| - | Maze is playable | 2020-12-14 |
| Username search | Fully functional search | 2020-12-14 |
| Username submission | Fully functional storage | 2020-12-14 |

Non-functional Requirements

|  |  |  |
| --- | --- | --- |
| Maze generation time | Maze should take less than 5 seconds to generate | 2020-12-16 |

Software Requirements

|  |  |  |
| --- | --- | --- |
| Browser testing | Works on all HTML5 browsers | 2020-12-16 |

## Tests for Functional Requirements

<Describe your test results for the functional requirements.

TODO: Provide a list of use cases or functions you have tested, as well as the testing results (whether or not the system passed the tests).

|  |  |  |
| --- | --- | --- |
| Maze generator | Maze is generated | Success |
| - | Maze is playable | Success |
| Username search | Fully functional search | Failure (maze seeds did not work in generator) |
| Username submission | Fully functional storage | Success |
| Browser testing | Works on all HTML5 browsers | Success |

## Tests for Non-functional Requirements

<Similar to the Section 4.2, but this section is for the non-functional requirements. >

|  |  |  |
| --- | --- | --- |
| Maze generation time | Maze should take less than 5 seconds to generate | Failure (Hard mazes took 15 seconds to generate) |

## Hardware and Software Requirements

<Describe the hardware and software requirements for performing the tests. >

Software Requirements

|  |  |  |
| --- | --- | --- |
| Browser testing | Works on all HTML5 browsers | 2020-12-16 |

# Analysis

<In this Section you need to analyze the effort that has been put on this project.

TODO: Describe how many hours (approximately) each team member spent on the project, for each milestone, which milestone took the most effort and why.

Daniyal:

Andrew: ~ 20 hours

Matthew: ~ 30 hours

Patric: N/A

Milestone 1

Milestone 2

Milestone 3

# Conclusion

<Conclude the document with what you have learned through working on the project.>

Appendix A - Group Log

< Describe how frequently the group members meet during the semester, and how effective the communication is. This is optional for one-person projects.>

Our group usually met at least once a week throughout the semester.

|  |  |
| --- | --- |
| 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 SRS 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.* |
| *2020-11-09 (6:30-6:45)* | *We created the main project repo and discussed implementation details.* |
| *2020-11-16 (3:00 - 4:00)* | *Delegation of tasks for the software design document. Each member was given a section to complete: Section 1 & 4 - Matthew, 2 - Andrew, 3 - Patric & Daniyal.* |
| *2020-11-30 (3:00 - 3:30)* | *We discussed possible implementation paths and set out exactly what we wanted the website to look like.* |
| *2020-12-07 (3:00 - 4:15)* | *We delegated tasks to certain members in the group and created a to-do list that broke down what was needed to complete the project into manageable steps.* |
| *2020-12-15*  *(10:20 – 10:30)* | *We went over what happened in our demo and discussed what tasks remained for all of us to complete.* |