

CS 320 Course Project - Software Design Document

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:** | **2020-11-21** |
|  |  |
|  |  |
|  |  |

Content

**Contents ii**

**1** **Introduction 1**

1.1 Project Overview 1

1.2 Definitions, Acronyms and Abbreviations 1

**2** **Activity Diagrams 2**

2.1 Search Username 2

2.2 Play Maze 3

2.2 Post Maze 4

**3** **Class Diagram 5**

3.1 Maze Application 5

**4** **Behavioral Diagrams 6**

4.1 Username Search Sequence Diagram 6

4.2 Maze Operation State Diagram 7

**Appendix A - Group Log 8**

# Introduction

## Project Overview

Our project is 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

|  |  |
| --- | --- |
| **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. |

# Activity Diagrams

## Search Username

Activity diagram for Search Username use case. This shows the process of the user entering a username query, processing that query, presenting results, and then deciding to play a maze that was saved under that username, or to return to the landing page.

## 

## Play Maze

Activity diagram for Play Maze use case. This shows the process of generating a maze based on selected difficulty, then the user playing through that maze while a timer runs or deciding to quit the maze and return to the difficulty selection screen. Once the maze is completed, the user is then transferred to the post maze screen.

## 

## Post Maze

Activity diagram for the post maze/ insert username use case. This shows the process of after completing a maze. The user will have the ability to replay the maze, return to the landing page, or enter a username in which to store the generation seed of the maze they just finished. If they do choose to save a username, they will then be able to replay the maze, or return to the landing page.

## 

# Class Diagram

## Maze Application

**A close up of text on a white background

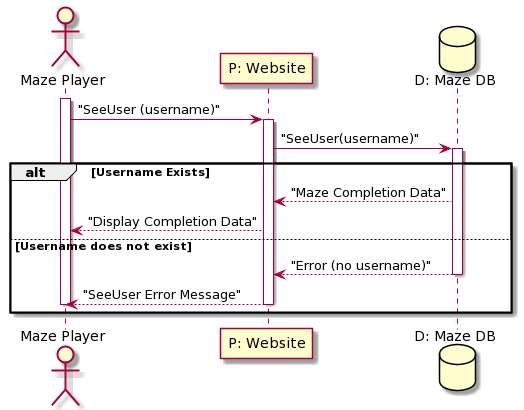
Description automatically generated**

|  |  |
| --- | --- |
| **Classes** | **Description** |
| DatabaseHandler | It get connection from database and access and retrieve data from database |
| LeaderBoard | It store information of users that are on the leaderboard |
| Score | It contains the time which is considered as score and a mazeText |
| Maze | It contains the 2d array of integer that is consider as maze and mazeText |
| User | It contains information related to user like score, username, mazes and its position on leaderboard |
| Game | It will be the main class that contains list of all users , leaderboard and reference to database. It will be the driver program of the game |

# Behavioral Diagrams

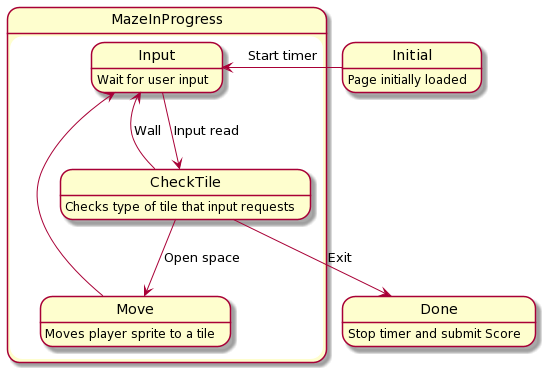
## Username Search Sequence Diagram

This diagram outlines the process of a user requesting maze data for a given username. The user submits the username and receives a response based on the existence of the username in the database.

**

## Maze Operation State Diagram

This diagram outlines the system that the maze will use to process user input. The maze enters a “MazeInProgress” state when the page is first loaded, checking for tiles when the user gives an input. Once the exit to the maze is found, the “MazeInProgress” state progresses to the “Done” state.

**

**Appendix A - 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 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.* |