

CS 320 Course Project Final Report

for

**Maze Web Application**

Prepared by

Group Name: *Team 19*

|  |  |  |
| --- | --- | --- |
| <name> | <student #> | <e-mail> |
| <name> | <student #> | <e-mail> |
| <name> | <student #> | <e-mail> |
| <name> | <student #> | <e-mail> |
| <name> | <student #> | <e-mail> |

|  |  |
| --- | --- |
|  |  |
| 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

*<TO DO: Please provide a brief introduction to your project.>*

## Project Overview

< A brief description of the project.

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

## 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.>

## References and Acknowledgments

<List any other documents or Web addresses to which this document 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 for this section.>

# 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

<Describe the development environment you were using for the project.

TO DO: List the programming languages, IDEs, tools, etc.>

## Task Distribution

*<Describ how the implementation tasks are distributed among team members.*

*TO DO: For each team member, describe his/her main implementation tasks in this project.*

*If this is a one-person project, mention: “all the work presented here is done by \*\*\* (your name).” >*

## Challenges

*<This section is optional. Describe the challenges in the implementation, if there are any, and how you dealt with them.*

*TO DO: If you don’t have anything to fill in, just leave this section blank.>*

# 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. >

## 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).>

## Tests for Non-functional Requirements

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

## Hardware and Software Requirements

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

# 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. >

# 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.* |