

CS 320 Course Project Final Report

for

Collaborator

Prepared by

Group Name: Team Collaborator

|  |  |  |
| --- | --- | --- |
| Brian Koga | 01152937 | brian.koga@wsu.edu |
| Kyle Trimbo | 11395503 | Kyle.trimbo@wsu.edu |

|  |  |
| --- | --- |
|  |  |
| Date: | December 12, 2019 |
|  |  |
|  |  |
|  |  |

Contents

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

Collaborator is a web application that helps find team members for software developers and hobbyists. It has features that cultivate this idea such as profile pages for users, project pages to help present ideas, as well as an advertisement section to help find projects or team members. This document details where the project is at after the requirements and design phases have been completed and the implementation phase has begun. This document includes information about the design and implementation of the project, as well as imformation about testing around the requirements.

## Project Overview

Collaborator is a web application with features that helps software hobbyists or developers find team members to complete projects. It can be difficult to find team members with differing skills and values to complete a team, so Collaborator attempts to fulfill this need. It purposefully aims for developers that are not looking for a job but rather projects that would interest them in their free time as not every project and team will be able to compensate them with monetary value.

The goal of the web application is to create a hub for software developers to find others who have similar interests and to be able to collaborate and finish software projects. However, since there are plenty of other websites focused on finding jobs in the industry, this is more focused on personal studies and presenting one’s work to find potential collaborators.

## 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.>

HTML: Hypertext Markup Language

CSS: Cascading Style Sheets

UI: User Interface

IDE: Integrated Development Environment

## 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

The main evelopment environment used in project was the IntelliJ IDE. As it is a web application some HTML and CSS was necessary, but the bulk of the application was built using the React JS library and the Semantic React UI library that works with it. Other libraries used include Lodash, which helps with the manipulation of objects and arrays;

TO DO: List the other libraries used.>

## Task Distribution

In general the workload was shared among the two group members, but Kyle focused more on the UI and visual elements while Brian focused more on the behavior of the application.

## Challenges

The biggest challenge faced was that neither member of the team had ever worked with the React JS library before. It is a very useful library and allows for the creation of dynamic webpages, but it can be a little confusing to new users and it took some time to become familiar with it.

# 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 takes the most effort and why. >

# Conclusion

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

Appendix A - Group Log

The group met every Tuesday and Thursday for several weeks, starting October 8th. After the second deliverable had been turned in and most of the details for the project were worked out, the majority of communication was virtual via text message or discord.