

CS 320 Course Project Final Report

for

JournalJay

Prepared by

Team Sizzle Snap

|  |  |  |
| --- | --- | --- |
| Laurel Anderson | 11638131 | laurel.o.anderson@wsu.edu |
| Irina Bejan | 11613022 | Irina.bejan@wsu.edu |

|  |  |
| --- | --- |
|  |  |
| Date: | 12/16/2020 |
|  |  |
|  |  |
|  |  |

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

*The project is JournalJay, a web-based journaling site where users will be able to make and keep track of journal entries. The user can also customize their entries. This is a digitalization of a traditional medium.*

## Project Overview

JournalJay is a web-based journaling site where users can make an account, sign into their account, make journal entries, review those journal entries, get help, or review their account information. This site will work on modern browsers such as Google Chrome and Mozilla FireFox. This convenient web journaling app will give more people the ability to air their thoughts in writing.

## Definitions, Acronyms and Abbreviations

Here are the definitions of terms used in this document.

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Web | A collection of websites stored in web servers. |
| App | Application. |
| Browser | A software application for accessing information on the Web. |

## References and Acknowledgments

[1] IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended

Practice for Software Requirements Specifications”, October 20, 1998.

# 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)”. >

The JournalJay system no longer has admins as one of the types of users. Therefore, figure 2 and figure 6 from the software design document is no longer valid.

## 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 programming languages used for this project are:

* HTML/CSS
* JavaScript.

The IDEs used by the team included:

* Intellij
* Visual Studio Code

Additional tools used for this project are:

* Miligram
* GitHub
* JavaScript general libraries

## Task Distribution

Both team members worked with each other to get all projects done. If one member was having trouble or if they did not have time to finish an aspect of their code, the other member would step in to help. Both members fully tested their own code then pushed their changes to GitHub for the other member to test. All issues found in testing were brought to the team member in charge of that section. Here is the general distribution of work:

Laurel Anderson – Login page, user account management, testing code, help page and account page.

Irina Bejan – Journal entry creation and customization, testing code, account page*.*

## Challenges

We were perfect.

# Testing

The code for this project was tested incrementally, meaning that the core components of the code were created and tested before the next components were implemented. All code was first tested by the person in charge of that code, and once that testing was done the code was pushed to GitHub and the other team member would test it.

## 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. >

Laurel oversees the user sign in page. Here are the things that she would like to test.

## 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.>

Tech side and

skills side.

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.>