easy news and updates system

Devon Dalrymple

Created: 2023-01-27

Last modified: 2023-01-27 15:48:00

Digital Portfolio

software Design journal (all-in-one)

Table of Contents

[1 Table of Figures ii](#_Toc125729025)

[2 Table of Tables iii](#_Toc125729026)

[3 Introduction 1](#_Toc125729027)

[3.1 About this Project 1](#_Toc125729028)

[3.2 Replacing the Old 1](#_Toc125729029)

[3.3 About this Document 1](#_Toc125729030)

[3.4 All-in-One Clarification 1](#_Toc125729031)

[3.5 Project Information 1](#_Toc125729032)

[3.5.1 Start Date 1](#_Toc125729033)

[3.5.2 Due Date 1](#_Toc125729034)

[3.5.3 Budget 2](#_Toc125729035)

[3.5.4 Project Requirements Clarification 2](#_Toc125729036)

[3.5.5 Project Risks 2](#_Toc125729037)

[4 Data Dictionary 3](#_Toc125729038)

[5 Competitor-Type Analysis 3](#_Toc125729039)

[6 Requirements 3](#_Toc125729040)

[7 Nice-to-Haves 3](#_Toc125729041)

[7.1 3](#_Toc125729042)

[8 Frontend Design 3](#_Toc125729043)

[9 Backend Design 3](#_Toc125729044)

[10 Database Design 3](#_Toc125729045)

[11 Deployment Design 3](#_Toc125729046)

[12 Test Plan 3](#_Toc125729047)

[13 Secrets Configuration 3](#_Toc125729048)

[14 Links/Guides for Reference 3](#_Toc125729049)

# Table of Figures

# Table of Tables

# Introduction

## About this Project

This project is to build a digital portfolio, befitting a back-end leaning entry-to-junior-level Web developer. Particularly this should show of some ASP.NET and Angular skills as these are the employment opportunities, I am most interested in. This has become more necessary now that graduation is coming and there is a lack of job offers.

## Replacing the Old

This system will replace a Google Web site that I made back in 2017. The previous site was made before I learned to program and is now embarrassing to have it still listed as my digital portfolio for a Web developer in 2023. The old is linked currently on LinkedIn and is potentially detrimental to how hirable I appear.

## About this Document

This document is a design journal, which means it includes various things that I have jotted down or diagrammed in order to design and implement the project. It will be made freely available on GitHub as a way to show off my design and technical writing skills when the digital portfolio goes online.

Since this is a personal project needing completion in a few months, with relatively little time available to it, many things such as a detailed set of CRC cards and class ADTs may not be included. It is not that I lack the knowledge to do them, but their gain in this small project and limited time availability make them less desirable.

## All-in-One Clarification

Normally, I would have requirements, competitive analysis, test planning, etc., in different documents with more details included. Instead, all of this will be lumped together with less detail into this design journal. This will make it easier to jump to different information, reduce duplication of information, and make the project simpler.

## Project Information

### Start Date

2023-01-27

### Due Date

2023-04-05

### Budget

The budget for this project is $150 for the first two years that this project is deployed, ignoring the cost of labor to develop, test, and deploy.

### Project Requirements Clarification

The planned budget for this project is $150 and it is meant to be used for hosting fees (if applicable), purchasing a domain name, acquiring SSL, buying domain protection, and potentially other items, such as an email with the domain name and cloud storage.

The due date is approximately one month from the date of finals completion. It is set this way to allow time in April to market myself towards prospecting employers. It is also this way since time in April will need to be focused on completing finals, especially the capstone project.

### Project Risks

#### Classwork is Overwhelming

It is quite possible with this being my final semester that classwork will become very difficult and may distract me from personal project development. In response, the end date has been set a month from graduation and a time block for 45 minutes from Monday to Friday has been allocated each week to work on this project and others.

If time is left after classwork is completed on schedule for a week, 2/3 of the backup time blocks will be allocated towards personal project development.

#### High Cost of Deployment

The budget of $150 may be too much for an extensive deployment. In response, the budget may be lowered by foregoing on niceties such as a domain email, using free SSL providers, and self-hosting the deployment.

#### Data Loss

It is possible that data loss could hinder the project’s completion. In response, each time work concludes on this project for the day, all changes will be pushed up to the develop branch on GitHub.

# Data Dictionary

# Competitor-Type Analysis

# Requirements

# Nice-to-Haves

## Labyrinth of Devon (v3.0)

### Why

The Labyrinth of Devon was my submission for my first and second semesters of college. It is based around the World of Zuul project from the BlueJ books. It is a simple text based game that in the second class when from pure command line to featuring some graphical elements as the text-based game was played.

To make this digital portfolio more interesting and as a semi-separate personal project, I want to remake a version of Labyrinth of Devon. The idea being that I could see how much that I have changed since my freshman year.

The previous version was built using Java and AWT/Swing. It featured a four-floor text-based dungeon experience. It had randomization of damage that took the min and max damage values from the weapon and character strength and reduced it by any defense modifiers. The player and enemies had a chance to dodge. The player could gain experience and level up. There were weapons, armor, and other items that could be equipped. Each turn allowed one command to be performed. If the enemies were not in the same room of the player on that floor of the dungeon, they could move to an adjacent room at random and surprise attack the player in the same round. The dungeon had three floors of enemies, items, and random room generation. On the final floor was what was meant to be a mini boss out of three choices. My theme to outdo the other students taking the course between two classes was to add randomization everywhere: four layouts, with a few rooms randomly selected for the floor’s list, and a random allotment of monsters and items. This gave my game, something that only one other student could boast of at the time, different experiences each time it was played.

However, the code was unoptimized, had boring lore elements, lacked depth, had a poor class system, and used heavy copy and paste. If I wanted 20 health potions for a floor, I had 20 lines of code do the adding of health potions. The GUI in the second version was mediocre and had no images for what occurred in text. The class system was meant to be Warrior, Rogue, Mage, and Peasant. The peasant class worked as expected since they were meant to be a challenge to do since they lacked bonuses. The mage was useless but stronger than the peasant since the magic system was never implemented. The rogue had no special abilities but was stronger than the mage and had better dodge chances. This led to an unintentional difficulty system based on class chosen (which due to time constraints, I left in because it was a “feature”).

How have I changed over the years? What could be done so much better, even with the time tacked on by making it Web friendly? This is a fun little experiment I want to do.

### Self-Applied Constraints

1. 24-hour limit, timed.
   1. If I take a short break for less than 30 minutes, the time will be left to run since I tend to continue thinking about a project on break
   2. If I stop for the day or long period, the time will stop
2. 4 hours will be dedicated to any initial set up of third-party libraries along with designing the game
3. 20 hours will partially be used for design but, mostly, it will be dedicated to implementing the game on the site
4. Third-party libraries are allowed but cannot be overused
5. It must stay true to the mostly text-based nature of the original two versions
6. It must be on its project page with a description of what was used and what it is about
7. It should feature at least five floors
8. It should have more lore elements
9. It should have a working-class system
10. It should have magic elements

### Prerequisites

This cannot be started until previous projects have been listed and detailed on other pages and 1/3 of the requirements for the portfolio are met.

# Frontend Design

# Backend Design

# Database Design

# Deployment Design

# Test Plan

# Secrets Configuration

## About

This section is for all of secrets needed in environment or secrets files for the application to downloaded onto a machine and run without issue. Only the key will be supplied for these.

# Links/Guides for Reference