# Software Development Report

## CS 3300-002 Spring 2025

Jack Dodge, West Manison, George Morales, Sebastian Sanchez

Table of Contents

[Software Development Report 1](#_Toc196673243)

[CS 3300-002 Spring 2025 1](#_Toc196673244)

[Introduction 2](#_Toc196673245)

[Development Process 3](#_Toc196673246)

[Methods 3](#_Toc196673247)

[Timetables 3](#_Toc196673248)

[Challenges 6](#_Toc196673249)

[Architecture and Design 7](#_Toc196673250)

[Software Architecture 7](#_Toc196673251)

[Software Design 7](#_Toc196673252)

[Metrics 9](#_Toc196673253)

[Languages 9](#_Toc196673254)

[Run Times 10](#_Toc196673255)

[Conclusion 10](#_Toc196673256)

## Introduction

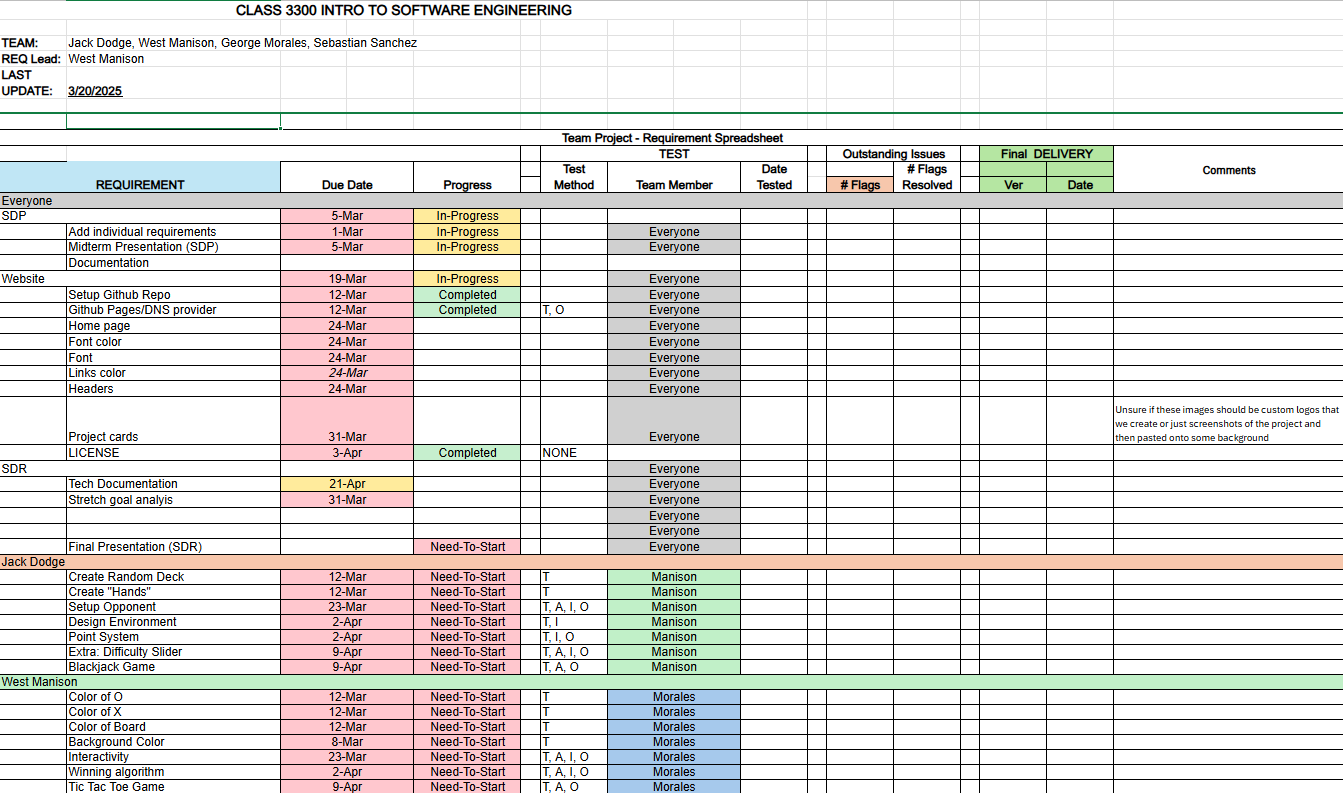
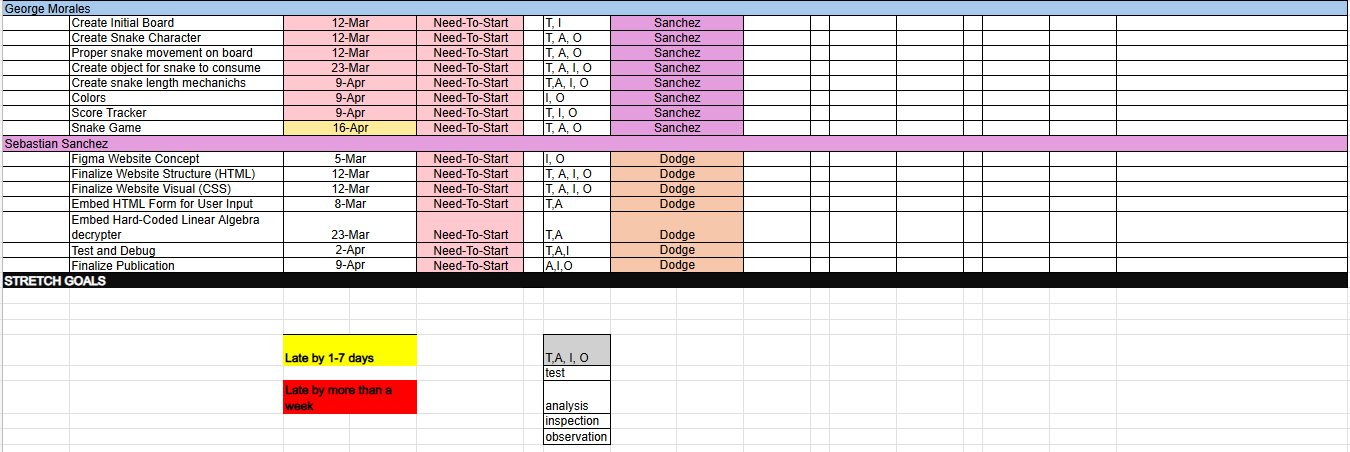
In the project, the goal was to create a portfolio website for personal use by the developers. The problem that we were trying to solve is the problem of standing out in a job market that is growing ever more competitive. The number of CS degree graduates and holders continues to rise in contemporary times, and the process of making oneself stand out to potential employers is starting to become less of an extracurricular and more of a requirement. A portfolio website can accomplish this while giving an employer a window into the life and work of an employee that they may be screening for a job. A portfolio website can contain biographical information, as well as built-in or links to projects that an employee has completed or participated in. In essence, a portfolio website allows a person to show off their personal skills to anybody who may be interested in more information about them.

## Development Process

### Methods

We first planned on using the Plan-Driven Development Process. However, this later developed into more of an agile-like approach. Some members who had more time than others were able to help those others out in order to complete the project in the set timeframe. We found agile to be much more in line with our working style, as everyone did what they could when they could, and any shortcomings were able to be fixed on the fly.

### Timetables



The screenshot above shows the original timetable used to dictate the schedule for this project. As can be seen, most of the original development was supposed to be completed in the month of March, with testing and fine-tuning set for the month of April. Unfortunately, the group was not able to stick to this ambitious schedule, and the following screenshot shows the true timeline that was used after the project was completed.

A spreadsheet with different colored lines

AI-generated content may be incorrect.

As can be seen in the contents of the screenshot above, the initial development schedule was not very inline with the true development schedule. Lots of the user code was completed within the final three weeks of the project deadline, as other college and life obligations got in the way of development.

Despite this crunched schedule, most of the things that needed completion were completed. Problems came in towards the end of the project, when it was realized that there would be some trouble when it came to making Python code work on static webpage applications. This meant that the original code developed for the Snake and Blackjack games had to be modified in order to make them compatible with GitHub Pages.

### Challenges

The Project was interesting in the challenges that were faced throughout the development cycle. Due to the wildly different life schedules that all of the groupmates had, the only times we were ever able to meet in person were right after the class sessions ended on Mondays and Wednesdays. However, this was heavily supplemented by the use of the messaging app Discord. The group was able to communicate effectively through the app’s messaging system, allowing for everyone's busy schedules to have a small impact on development. Development of the project started off very fast, as the main homepage for the portfolio site was created within the first two weeks of development. After that, development slowed. Most of the project code was developed within the last month of the project’s due date. Most of the documentation was developed as it was needed.

We originally planned to try and make this website so that any user could come in and make their own page on the website, but we soon realized that this was going to require an entire revamp of what we already had. We realized that we would not have enough time within our deadline in order to construct this social media-like website, and decided to push forward with hosting the page on a static webpage application. One of our future/stretch goals with this project is to eventually make it so that user accounts can be created, allowing for anybody who is interested to come in and make a page of the website that they can use to showcase themselves.

## Architecture and Design

### Software Architecture

The following image is an image of the software architecture for this project.

A diagram of a computer network

AI-generated content may be incorrect.

As can be seen, the design of this project is very simple. Since the project is being hosted by either GitHub Pages (or a user’s local client, if they prefer to download the repository and run the code on their own), the website is static, and there is no complex backend that needs to be compiled. The static page is served over the Internet to whatever device is requesting it.

### Software Design

The following screenshot shows the software design for this project.

A diagram of a website

AI-generated content may be incorrect.

As can be seen, the software diagram is relatively simple. A user would start on the homepage of the document, and would then be presented with options to contact John Doe, view John Doe’s social media profiles, such as GitHub and LinkedIn, see John Doe’s portfolio, or see one of the projects on the website. If the user selects one of the projects, they will be presented with a page that is solely dedicated to the respective project. Each project page will be served up by HTML with the project and its respective file/language embedded into the page. Snake and Blackjack were created in Python, while Tic Tac Toe and Hill Cipher were created in JavaScript. If the user so chooses, they can also see the source code for each of the files for that specific project. Selecting one will bring them to a dedicated page that has the source code laid out on an HTML page to view. As can be seen in the top right corner of the diagram and the line that leads through all of the different sections of the diagram, CSS and site images remain a constant part of every page on the website.

➢ One report on entire development activity  
➢ One final section is Software Metrics (line count, run time, file naming convention, etc) – this paragraph also is included in User’s Manual and Developer Documentation

## Metrics

### Languages

As of the latest update to this document (April 27th, 2025), there are currently 1995 lines of true code in the entire repository. The following screenshot is a ‘cloc .’ command run on a local version of the repository.

A screenshot of a computer

AI-generated content may be incorrect.

This shows the number of files in the repository, the number of blank lines, comment lines, and code lines. In this case, there are 25 files in the repository, 480 blank lines, 482 comment lines, and 1995 lines of actual code in the repository. The naming convention that was used for this entire project was Camel Case.

### Language Breakdown

The language breakdown of the repository can be seen below.

A screen shot of a computer

AI-generated content may be incorrect.

As can be seen, the majority of the code in the repository is HTML code, reflecting the need for the Hypertext Markup Language in the development of a portfolio website.

### Run Times

The following are page metrics obtained from running a scan on the website “webpagetest.org”.

A screenshot of a graph

AI-generated content may be incorrect.

If you would like a detailed output of all of the tests run from this website, it can be found in the public repository labeled “Runtime & Lines of Code.csv”. You can also click this link [here](https://github.com/WManison/WManison.github.io/blob/main/docs/Misc/Runtime%20%26%20Lines%20of%20Code.csv) to take you to this page.

## Conclusion

Overall, this project served as an important learning experience for all of the members involved. This was every member’s first time working on a big collaborative GitHub project on a strict deadline, and many good life lessons were learned along the way. While lofty goals were set, many things were modified and adjusted, attributed mostly to life circumstances and a finite timeline. Nevertheless, the majority of this project was completed, with only a few non-functional requirements not being achieved. We also have stretch goals in mind for future project development. We hope to apply all of the things we learned with this project to any future endeavors that we may encounter.