|  |
| --- |
|  |
| CSS 497 Capstone Portfolio  **Project**: A Virtual Tabletop for Game Masters  **Prepared for**: Game Masters and Players  **Influenced by**: John Yeung – Assoc. Dir. of Infrastructure Services  **Created by**:Jasper Yeung  **Quarter**: Summer 2025 |
|  |

Table of Contents

[Part 1 – Reflect on your capstone learning 3](#_Toc206770464)

[Devising the Project 3](#_Toc206770465)

[Planning the Project 3](#_Toc206770466)

[Designing the Project 4](#_Toc206770467)

[Gathering Feedback 4](#_Toc206770468)

[Project Implementation & Progress 5](#_Toc206770469)

[Learning New Tools 5](#_Toc206770470)

[Status Reports and Hour Logging 6](#_Toc206770471)

[Part 2 – Reflect on your status reports 6](#_Toc206770472)

[Scheduling & Progress 6](#_Toc206770473)

[Reporting Accomplishments & Feedback 7](#_Toc206770474)

[Overall Insights 8](#_Toc206770475)

[Part 3 – Reflect on your meetings 8](#_Toc206770476)

[Introduction 8](#_Toc206770477)

[Librarian 8](#_Toc206770478)

[Instructor 9](#_Toc206770479)

[Stakeholders 9](#_Toc206770480)

[Expert 9](#_Toc206770481)

[Part 4 – Reflect on your career prep activities 10](#_Toc206770482)

[Introduction 10](#_Toc206770483)

[Self-Assigned Career Center Activity 10](#_Toc206770484)

[Create Questions for Interviews Activity 11](#_Toc206770485)

[Update Husky Handshake Profile 11](#_Toc206770486)

[Conclusion 11](#_Toc206770487)

[Part 5 – Capstone Artifacts 12](#_Toc206770488)

[Documents 12](#_Toc206770489)

[Architecture Diagrams 14](#_Toc206770490)

[User Experience Diagrams 16](#_Toc206770491)

[Part 6 – Annotated Bibliography 19](#_Toc206770492)

[TTRPG Sources 20](#_Toc206770493)

[Tools and Methodologies Research 20](#_Toc206770494)

[Tool Documentation 21](#_Toc206770495)

[Tutorials 21](#_Toc206770496)

[TypeScript and React Setup 21](#_Toc206770497)

[TypeScript and React Basics 22](#_Toc206770498)

[Feature Implementation 22](#_Toc206770499)

Part 1 – Reflect on your capstone learning

Devising the Project

This virtual tabletop (VTT) capstone project was conceived to accomplish two main design goals: to support multiple tabletop roleplaying game (TTRPG) systems and providing useful features that are applicable to either individual or multiple TTRPG systems. These two goals were identified because of my own experience as a GM and the perceived problems I have found with VTTs on the market. For example, the main VTT I use is AboveVTT which is extremely feature rich and useful but only works for Dungeons and Dragons 5th Edition (D&D 5E). On the opposite spectrum is Roll20, a virtual tabletop that supports numerous TTRPG systems but has very general features that are not super useful.

From the GM’s perspective, the main draw of a VTT are the convenient and useful features that reduce preparation materials and accelerate cumbersome rule arbitration and calculations. But because TTRPG are very rules heavy, general features are not as helpful to have. But GMs would also like to have a single VTT that supports multiple systems with familiar UI and user experience. Furthermore, there are many TTRPG’s on the market that cannot have their own dedicated VTT unlike more popular games like D&D 5E and therefore do not have useful and convenient features to use. This VTT capstone project aims to solve all of these wants and needs by creating software architecture that can support multiple TTRPG systems with modular features able to be added and removed from each system.

Planning the Project

Subsequentially, devising the project’s concept and design requires a foundation to determine what features should be implemented and prioritized in order to accomplish these goals. That foundation for my project was creating comprehensive software requirements and specification (SRS) documents. In the document I organized the functional requirements under the different overarching features being implemented while providing their development priority and additional non-functional requirements. The development priority of these features considered the importance of the feature to a GM, whether the minimum viable product requires its completion, and it’s estimated difficulty to implement.

To provide an example, the main feature of the project is the Battle Map feature which had 13 total functional requirements and a priority of high due to its completion being required for a minimum viable product as the most basic and useful feature in a VTT. In total, the SRS document contained 65 functional requirements over 10 features and 7 non-functional requirements. This document was not only useful for solidifying the ideas I had for the project, but also for planning schedules and creating diagrams.

Due to the creation of the SRS document, the planning of the schedule could be created. The process of planning the schedule considered the required capstone requirements, creation of design artifacts, development of features based on priority, and testing plans. Because CSS497 provides a clear schedule of assignment deadlines and meetings, it was simple to add them onto the schedule with their estimated completion time. This was especially the case for creating the status reports which I aimed to complete for all nine weeks where I simply used the time I took for the first status report as a baseline estimate for the subsequent status reports. I then planned for the interactions between stakeholders, expert, instructor, and librarian meetings which were more flexible but were best done at specific time intervals. All of these meetings needed to be done early on to gather feedback and information needed to improve initial plans to then implementation. Then the meetings requiring more than one interaction were scheduled towards the end of the summer quarter, so enough artifacts and features were completed to gather additional feedback.

The last considerations were the feature implementation and testing plans. These were used to fill in any remaining time each week in order to reach 20 total weekly hours. Each feature would be scheduled based on an initial estimation of the development time needed with the expectation that high priority features may push back lower priority features in the schedule. The unit testing plans were scheduled after each feature was planned to be finished and functional testing plans scheduled for after enough features were implemented to have basic functionality.

Designing the Project

In addition, the next phase of the project was to create multiple diagrams in order to design the software architecture and user interface. During this phase, a domain, class, component, and flowchart diagrams were created along with a mockup of what the web app would look like. The domain diagram was the first diagram to be created which connected the relationship between various TTRPG objects and terms that I may have to consider while implementing the project. Next, the class and component diagrams were the diagrams that were used to plan for how the project implementation would be organized after considering the domain diagram objects. These diagrams were especially important because they are the fundamental building blocks to create a software architecture that supports multiple TTRPG systems with useful features. On the other hand, the flowchart diagram and mockups were created to design what the user experience would feel and look like. The flowchart focused on the overall user experience and how they would use and exit from the web app while the mockup focused on what the UI would look like.

Gathering Feedback

Notably, the mockup was created in response to feedback from the expert meeting looking over the artifacts previously mentioned which were mostly compiled into a design specification document. In the meeting, the expert gave significant feedback regarding localization considerations, development tools to use, and testing plans. Because the project is a web app, the expert provided feedback that localization considerations are important for the layout of the UI and image processing of the project. The expert also suggested that additional development tools should be used or experimented with such as Azure DevOps to streamline metrics, testing, and deployment. Finally, the expert gave feedback in regard to my initial testing plans.

My initial testing plans were divided into unit testing, system testing, integration testing, and usability testing. However, the expert pointed out that my understanding of system testing where certain unusual behaviors are tested for issues is functional testing while system testing is about testing performance and resources. I also learned from the expert that I should have a better testing plan especially about having a testing framework and a testing environment. In response to this feedback, I revised my testing plans to use Jest as the unit testing framework while learning how to use Azure DevOps.

The last activity performed before implementing the project was the stakeholder meeting. For my stakeholder meeting I interviewed another GM to ask about their GM experiences, their familiarity with VTTs, and features they expect and desire. Because I am also a GM, it would be easy to be biased in my own experiences as a GM so meeting with a GM with different experiences is a great way to gather feedback for what I should prioritize or add to the project. One such important piece of information was other roleplaying games. While I focused only on TTRPG games like D&D 5E and Call of Cthulhu, I did not consider other roleplaying games that could also benefit from VTT such as Realming and even larping. I also did not consider that a GM would prioritize auditory features over visual features to better immerse their players. This feedback made me reconsider the audio feature’s development priority to be at a medium. This experience truly showed me that stakeholder meetings are extremely important to better understand as widely as possible the demographics and needs of your users in order to fulfill their needs.

Project Implementation & Progress

Consequently, after finishing the design phase of the project, the schedule and SRS document helped monitor the progress of my project which helped reflect on the leaps in progress and setbacks that I encountered. Each week I would complete all the scheduled work on the quarter schedule while prioritizing capstone requirements over project development. In doing so over the quarter, I have realized that the initial schedule was too optimistic to the realities of development. The features even considering development priority I allocated at three weeks max to complete. However, just working on the Battle Map and Token features has taken 5 weeks and more to be expected. In that time only 7 functional requirements were completed and 6 functional requirements partially implemented. This made me realize that the prototype at the end of this capstone project will have significantly less features than I wrote in the SRS document. That in itself is not detrimental but the fact that I have not been able to complete the two most important features and haven’t started the two other high priority features is concerning.

Learning New Tools

However, a significant part of why there were delays in feature implementation was learning the new tools and looking at their documentation. For this project, I had three main tools that were new to me and integral to the project: Fabric.js a library to create interactive web app, TypeScript with React the base programming language in the framework to implement the project, and Chakra UI the library used for implementing UI components into a web app. There were multiple instances in which, I was using these tools to implement a feature and an issues came up where I need to parse through documentation and other sources to find a solution to a problem I encountered. However, it took a significant amount of time just to find a hint of a solution which did not help when documentation was inadequate for my issue.

To illustrate this, I was starting to implement a sidebar menu in which there would be multiple tabs with different functionality for Tokens, Battle Maps, Roleplaying Scenes, and Audio. However, the basic requirement is for the sidebar to be displayed on the screen while allowing the Battle Map itself to be able to interact with Tokens or pressing buttons on the toolbar. With Chakra UI, the perfect looking component for the sidebar menu was the Drawer component. Unfortunately, the Drawer component did not allow any interaction with elements outside of the component itself. Therefore, the component could not be used as it’s default. While looking in the documentation for a solution I found a parameter called modal which in documentation stated it would allow interaction outside of the drawer. However, changing the parameter still did not allow outside interactions. After looking through various other sources, I found another Chakra UI component called the Action Bar which required more CSS to look like a sidebar but had the basic functionality needed of allowing outside interaction.

This illustration shows that seemingly perfect solutions must be exchanged with not ideal but functional solutions to be used during development. This is an inevitable process of development that was exacerbated to require even more time because I am still learning these tools. However, as I get increasingly familiar with these tools the time it takes to find functional solutions will also decrease.

Status Reports and Hour Logging

Furthermore, a large part of the development process was consolidating and writing down the progress of the project in the status reports. For the status reports, I always made sure to allocate time every Saturday to fill them out. The status reports were important artifacts to showcase my actual schedule and progress completed compared to my original schedule. I am usually routine person who follows a schedule each week for my assignments, so I had little difficulty finding time to complete the status reports. A notable change was observed throughout the quarter where during the design phase I allocated a part of the 20 hours to complete the status report but for the implementation phase I went over 20 hours each week. What I learned was that when I focused on a single task in a day, I would be better motivated to complete those tasks and the necessary amount of time needed to be spent especially during development. Finishing the status reports on a Saturday not only allowed me to better write down everything I accomplished each week but also helped me better focus finishing tasks during the week.

Part 2 – Reflect on your status reports

Scheduling & Progress

Throughout the project’s development, I made it my goal to create status reports for every week of the quarter. The status reports not only allow the instructor and experts to gauge the progress of the project but also allows me to compare my progress with the original schedule. In doing a status report for each week, I have realized that my progress on capstone requirements have all been mostly the same while the implementation and testing of features were significantly different than originally planned.

The original schedule was planned around the priority of capstone requirements, meetings, then feature implementation and testing. The capstone requirements were scheduled to follow the provided schedule of CSS497 for the summer quarter with drafts being completed the week before the due date. The meetings were scheduled depending on the other participants’ availability but were scheduled to be towards the beginning of the quarter or end of the quarter where their input was most useful and enough artifacts would be developed to be shown. As for, feature implementation and testing, they were scheduled based on the priorities determined in the SRS document based on the number of functional requirements, the importance of the feature, whether the feature was required in the minimum viable product, and the perceived difficulty of the feature’s implementation. Through this criterion, the schedule was created around the capstone requirements with meetings flexible to be changed and feature implementation and testing filling out the remaining hours needed to complete 20 hours each week.

Throughout the process, however, the actual schedule in the status reports had capstone requirements and meetings staying mostly to schedule while the implementation and testing of features were significantly altered. The main reason for this divergence is that the Token and Battle Map implementation has taken significantly longer than anticipated. The reason for this gap in expectation has to do with my experience using Fabric.js and Chakra UI. During the schedule’s creation I only chose the tool and libraries I would be using and only learned surface level details of how to use them. As a result, significant time had to be spent reading documentation and finding solutions while learning the tools at the same time. Because of these considerations, each week my schedule plan for the next week would be based on the original schedule while modifying the feature implementation and testing activities.

However, that is not to say that no work was being done. While only 7 functional requirements were completed and 6 partially implemented, these functions are fundamental to the project. A significant number of requirements in the SRS document using MoSCoW notation were categorized as “should” or lower. But the requirements implemented so far have been “must” requirements which are significantly more important. This has been reflected in the accomplishments and evidence of work presented throughout many status reports.

Reporting Accomplishments & Feedback

The accomplishments of the status reports are shown in a combination of artifact completion states, information gained from activities, and features implemented. The artifact completion states were kept track of based on how much of the artifact was complete in terms of prompts and instructions such as the design specification and capstone contract documents. Non-required artifacts such as the mockups were documented with brief explanations of their purpose and state of completion. For information gained from activities, these were tracked by the notes I wrote from my activities such as research which were presented as evidence of work. The summary of the information found would then be used as an accomplishment for a status report. Finally, the features implemented were shown through screenshots of the feature working while written as what functional requirement was implemented.

The feedback I received from each week whether through meetings or showcases were written in the format of the person who gave the piece of feedback and my response to their feedback. For some weeks, I would have at most one or two pieces of feedback each from a different person but for the expert meeting I had many. Due to how much feedback I received on week 4, I summarized the most important feedback I received in separate sections based on their topics.

Regarding responding to feedback, during the first week I did not write notes on who gave me feedback. As such, I had significant trouble remembering who gave the pieces of feedback that I remembered receiving and had to guess. This experience made me realize that I should take as many notes as I can when receiving feedback and attribute it to the correct person.

Overall Insights

Finally, through the reflections made throughout the quarter and all the status reports, I have found that a significant portion of my reflections were responding to and solving development issues. The schedule is one aspect of development issues, but others include reading through lacking documentation and finding workarounds to problems. Fabric.js despite its useful functionality, has a documentation website that has many pages missing and many parameters and functions lacking in explanation. Chakra UI on the other hand has clear documentation for common use cases and issues but requires lots of additional research for unorthodox functionality needed for my project. Despite all these problems though, the status reports show that I have overcome these problems even if the schedule had to be modified. Adapting to these changes and issues is a necessary skill for a software engineer that I have honed by working on this project.

**View status reports**: <https://github.com/Jyeung29/VTTWebApp/tree/main/Project_Artifacts/Status_Reports>

Part 3 – Reflect on your meetings

Introduction

Throughout the quarter I met with various individuals in different roles which helped inform the development of this project. I met with the librarian, instructor, stakeholders, and an expert in order to gather feedback and information from their different roles and perspectives.

Librarian

I met with the librarian Je Salvador during week 4 of the summer quarter. I would have liked to schedule earlier with the librarian, but various scheduling conflicts pushed the meeting back. Fortunately, during week 4 the project was nearing the end of the design phase and could still yield useful information from the librarian. During the meeting Je gave me many resources I could use including databases, search methodologies, public APIs, and even game benchmarks. This meeting provided extra resources that I could use if I needed additional sources to implement my project.

Due to the meeting taking place near the end of the design process, I already researched the tools I would be using for the project so I have not ended up using the public APIs Je provided. What I have primarily gained from the meeting was the search methodologies and the web design resources. I have had to do significant research for fixing issues and finding the right tools to implement the project. The searching methodologies of keywords and reliable sources is a general skill that is useful no matter what stage of the project I am in. As for the web design resources, I am primarily using Chakra UI for web design components and have focused on functionality rather than appearance. While I am not currently reading the sources provided, I will definitely do so once my project is in a state where I can adjust the web design to be more user friendly.

Instructor

The instructor meetings with Professor Dawn-Marie Oliver were conducted three times throughout the quarter to check in on progress and ask questions. These meetings were usually short in length, around 10 to 20 minutes. This is because the questions I asked were generally short and mostly clarifications for assignments. Because I am a quarter 1 student, I did not have to ask the professor questions about the capstone posters and presentations. While the capstone design specification, capstone contract, and capstone portfolio were all long writing assignments, I consider myself a generally strong writer and the professor evaluated my writing as meeting a high standard with mostly minor changes needed.

The other questions I asked were regarding how stakeholders and experts needed to indicate their interaction with me and whether certain activities such as the capstone colloquium attendance should be logged in the status report. These questions then helped me better understand how to complete their associated assignments and plan my schedule.

Stakeholders

The stakeholder interaction was held with Ariel Lim with experience as a game master. Ariel as a stakeholder was valuable because she is a different type of game master from me. During the meeting I asked about her experience with TTRPGs, her game master style, what tools she uses, what features she expects, and what features she desires. From the interaction, Ariel focused more on immersion and storytelling rather than rules heavy activities such as battle maps. She also did not have much experience with TTRPG tools such as virtual tabletops. This is in comparison to me who is a game master more focused on rules, heavy battles and has experience with other TTRPG tools.

The answer she gave for features was most valuable for the development of the project. The first was that given the choice between a website and a desktop app for a virtual tabletop, she would choose the website. This feedback reinforced my choice to make this project a web application. Second, the features she expected such as battle maps and tokens were standard. But she emphasized that she would prioritize audio features such as music over visual features. This feedback made me reconsider the low priority I placed onto the audio controller feature. Third, the features she desired were connected to Spotify, DND Beyond, and full body character images able to be displayed. The DND beyond implementation and full body character images were already features that I outlined in my SRS document. However, the connection to Spotify was something I overlooked, and I believe would make an excellent feature.

Expert

The expert interaction was held with John Yeung who has experience with web application development during his career. During the two interactions various artifacts were reviewed and a code review was performed. The main feedback during the first meeting was regarding testing plans, localization considerations, and development tools to assist in testing, source control, and release management. This meeting allowed me to better refine my testing plans to consider different screen resolutions, device types, and languages. It also guided me on possible tools I could use such as Azure DevOps to assist in development. The second interaction reviewed the mockups, testing logs, and source code. The main piece of feedback received was the use of a single file for configurations, localization text, and CSS or component styling. The current project iteration has multiple CSS files, in-line styling, and hard-coded references to images. John noted that it is best practice to refactor the code in order to have a single reference file for each of those purposes. This feedback better helped me understand how in professional environments, it is best practice to have a centralized file for a purpose in order for release builds and testing builds to be easily configurable.

**Meeting reports**: <https://github.com/Jyeung29/VTTWebApp/tree/main/Project_Artifacts/Status_Reports>

* Status Report Week 1
* Status Report Week 3
* Status Report Week 4
* Status Report Week 6
* Status Report Week 9

Part 4 – Reflect on your career prep activities

Introduction

Throughout the quarter I completed three activities that would prepare me for my future career as a software engineer. These activities were chosen based on time constraints to fit within the capstone project schedule and how helpful they would be to complete.

Self-Assigned Career Center Activity

The first activity completed was the “Self-assigned Career Center Activity.” This activity was open ended where I could choose which career service area I would like to learn about. Because of my previous experience applying to internships and part-time jobs, I already learned how to do resumes, cover letters, networking, and interviews. Of the areas, I was most unfamiliar with references and letters of recommendation due to job applications not requiring them and even my college applications not especially needing them.

In this assignment I learned about who to ask for references, what you can provide to the reference person to help write a letter, and basic courtesy you can provide to the reference person. The main piece of information I did not know previously and was not easily assumed was that you should contact the reference person of the results of the application they wrote the letter for. Even if it was a rejection, the reference person should be notified that you used their reference, and you can thank them for helping you.

Create Questions for Interviews Activity

The second activity completed was the “Create Questions for Interviews” assignment. This assignment had me brainstorm questions that I may ask interviewers during an interview to better know their organization. In the assignment I used the career service’s example questions as inspiration to come up with new questions to ask regarding the organization’s values, team and work environment, and what it takes to succeed and organize them depending on the interviewer’s role.

For this assignment, I learned that there are many questions that I can learn that have to be targeted at specific interviewer roles. Depending on the company, interviewers may compose a panel of employees from different departments or only from a single department or team. My own experience during interviews was having a panel from only engineering team roles. I learned that it is important to ask the right questions to show your preparation and situational awareness of your and their position in the company and show your enthusiasm for the role.

Update Husky Handshake Profile

The final activity completed was the “Update Husky Handshake Profile” assignment. In this assignment I showed my previous Husky Handshake Profile and the new updates I made to the profile. The main changes I made were updating the summary to include mentioning my desire for full-time positions and my experience in frontend and backend tools. I also updated my skills, courses, and projects I have done since the last time I updated the profile.

In this activity I did not learn new skills but had the chance to refine and update my profile to better align with my current career status. As I am in my senior year, I must start looking and preparing for full-time software engineering positions. My summary had to be changed in order to reflect this and indicate to employers of skills I have learned and my willingness to go full-time.

Conclusion

After completing all three career preparation activities I feel that I am more ready for my future career than I was before. I learned new skills that I previously did not know about with references and letters of recommendation. I brainstormed interview questions that I could ask interviewers of various roles to show my interest and attentiveness. I updated my Handshake profile in order to show to and impress prospective employers of my skills and willingness to work full-time. Of course, my other skills in regard to resumes, interviews, cover letters and more are always going to need to be improved throughout my career. But I feel that after completing these assignments I will be able to better apply for new career opportunities.

**Career Prep Reports:**

https://github.com/Jyeung29/VTTWebApp/tree/main/Project\_Artifacts/Career%20Prep

* Career Center Activity Reflection
* Questions for Interviews
* Husky Handshake Profile Update

Part 5 – Capstone Artifacts

Documents

The Design Specifications document as seen in *Figure 1*, provides comprehensive explanation of the project and its design. This is accomplished through the various sections that expand upon the project’s basic information, resources used, detailed specifications, schedule, and contingency plan.

**Link:** <https://github.com/Jyeung29/VTTWebApp/blob/main/Project_Artifacts/Documents/Yeung497_designDocFinalDraft.docx>

A screenshot of a computer

AI-generated content may be incorrect.

Figure : Design Specifications Document

The Software Requirements and Specifications (SRS) document as seen in *Figure 2*, outline the functional and non-functional requirements of the project. The functional requirements are organized under the features they are associated with each with their own development priority assigned. All requirements use the MoSCoW format which outlines the priority of individual requirements.

**Link:** <https://github.com/Jyeung29/VTTWebApp/blob/main/Project_Artifacts/Documents/YeungSRS.docx>

A screenshot of a computer program

AI-generated content may be incorrect.

Figure : Software Requirements and Specifications Document

Architecture Diagrams

The Domain Diagram seen in *Figure 3*, provides a high-level view of what objects and concepts relate to the project’s domain. This domain diagram focuses on showing the relationship between TTRPG concepts, especially regarding common rules between TTRPG systems. The Domain Diagram helps inform the creation of future architecture diagrams such as the Class Diagram.

**Link:** <https://github.com/Jyeung29/VTTWebApp/blob/main/Project_Artifacts/Diagrams/vttDomainDiagram.drawio.png>

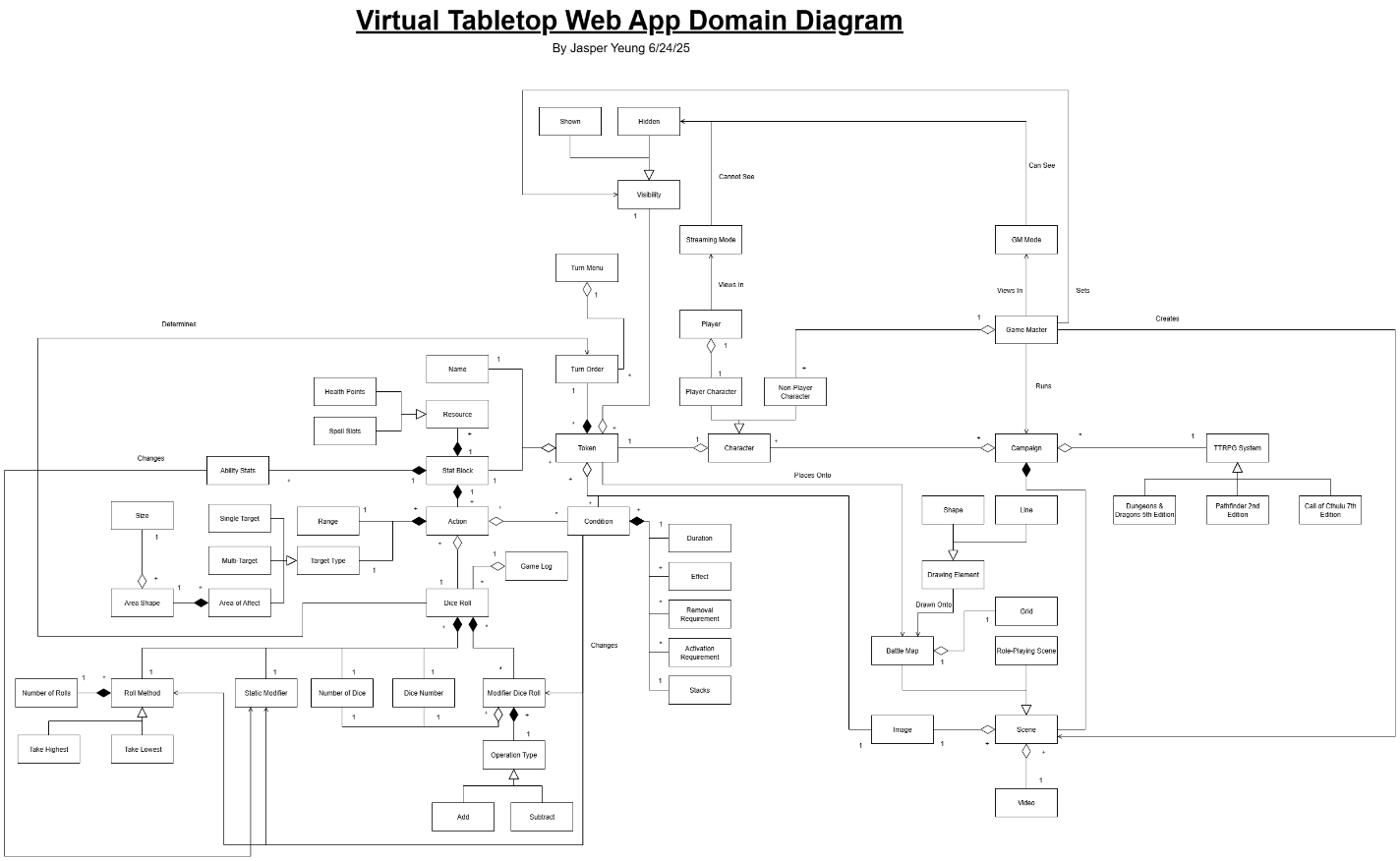


Figure : Domain Diagram

The Class Diagram presented in *Figure 4*, provides a low-level view of classes that may need to be implemented for the project. The diagram shows classes with their class members and functions. Many of the classes in the diagram have composition relationships to other classes which coalesce to the Campaign class. The TTRPG in particular is an abstract class with child classes able to have different features which help create a software architecture supporting multiple TTRPG systems.

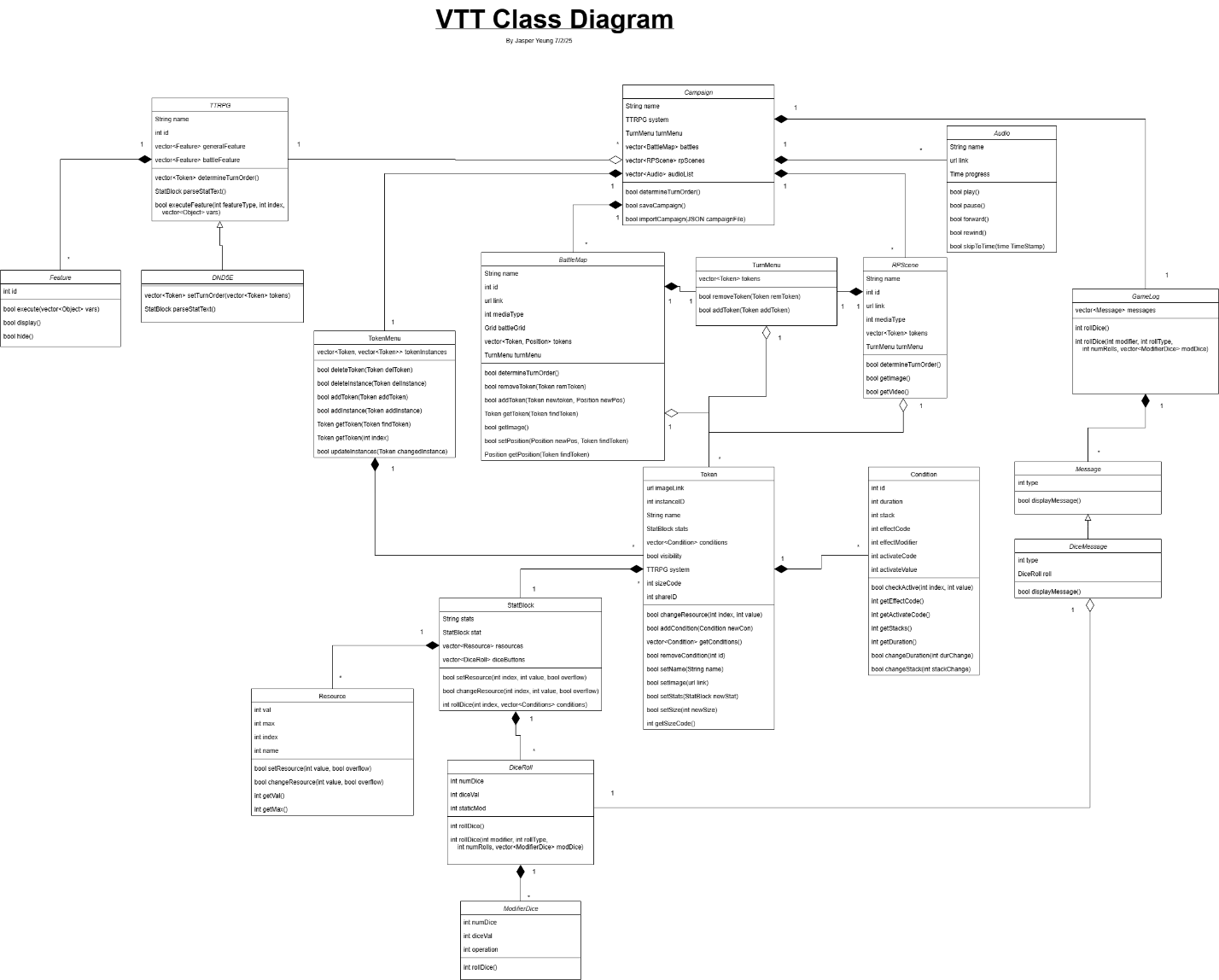
**Link:** https://github.com/Jyeung29/VTTWebApp/blob/main/Project\_Artifacts/Diagrams/vttClassDiagram.drawio.png  


Figure : Class Diagram

The Component Diagram as seen in *Figure 5*, provides a higher-level view of the various components that need to be implemented for the project. The diagram is organized in three distinct groups with representation logic, campaign elements, and tokens. Many components have aggregate or composite relationships while some have interface relationships in order to implement proper separation of concerns.

**Link:** <https://github.com/Jyeung29/VTTWebApp/blob/main/Project_Artifacts/Diagrams/vttComponentDiagram.drawio.png>

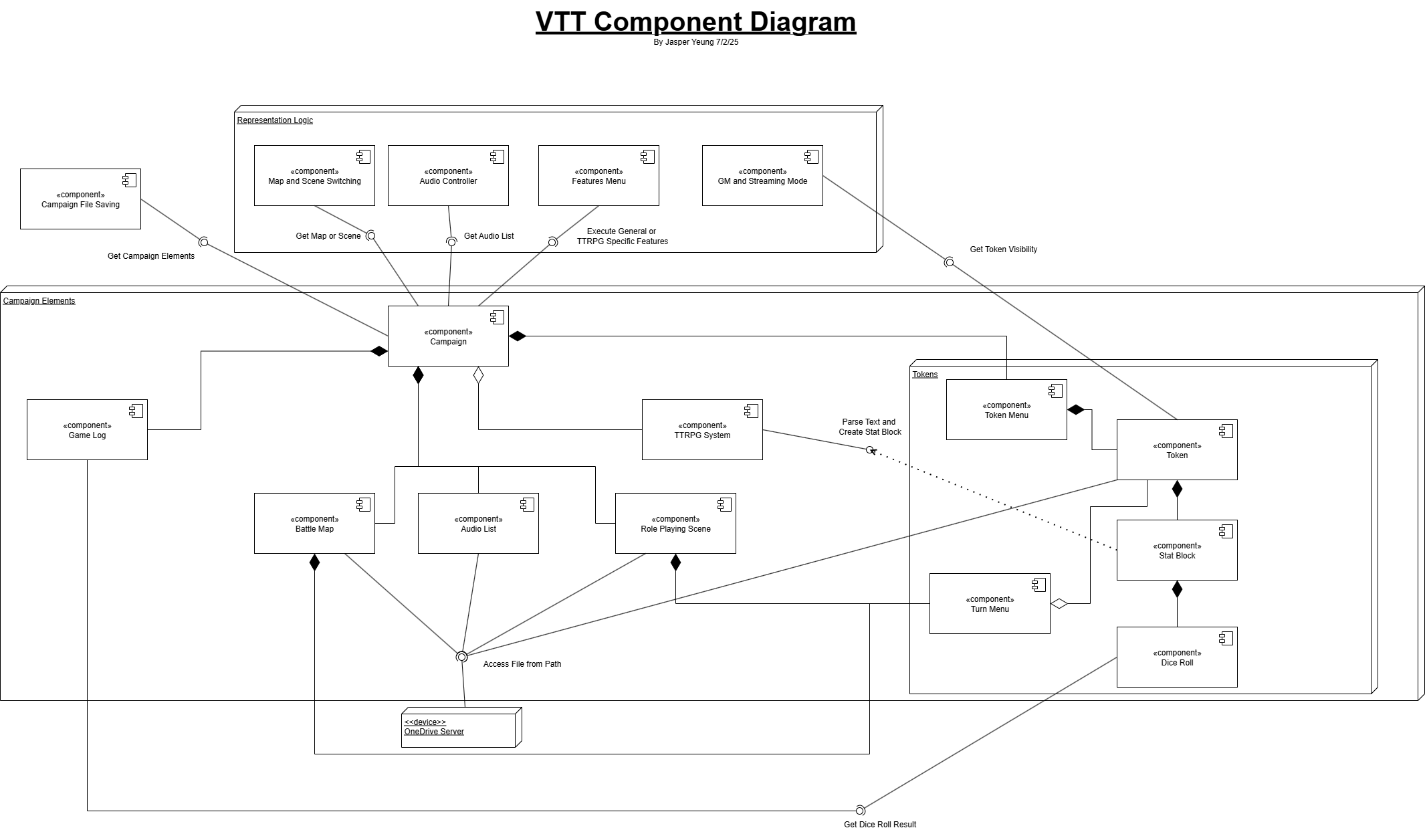


Figure : Component Diagram

User Experience Diagrams

The Flowchart shown in *Figure 6*, provides a high-level view of a GM’s user experience when using the VTT. The Flowchart starts when the GM opens the web app to create a new campaign or upload a previous campaign file. The middle section is where the main functionality of the project is used such as Battle Maps, Tokens, and Roleplaying Scenes. The last section is where the user wants to exit from the web app where some prompts may appear based on if the user has saved their campaign.

**Link:** https://github.com/Jyeung29/VTTWebApp/blob/main/Project\_Artifacts/Diagrams/vttFlowchartDiagram.drawio.png

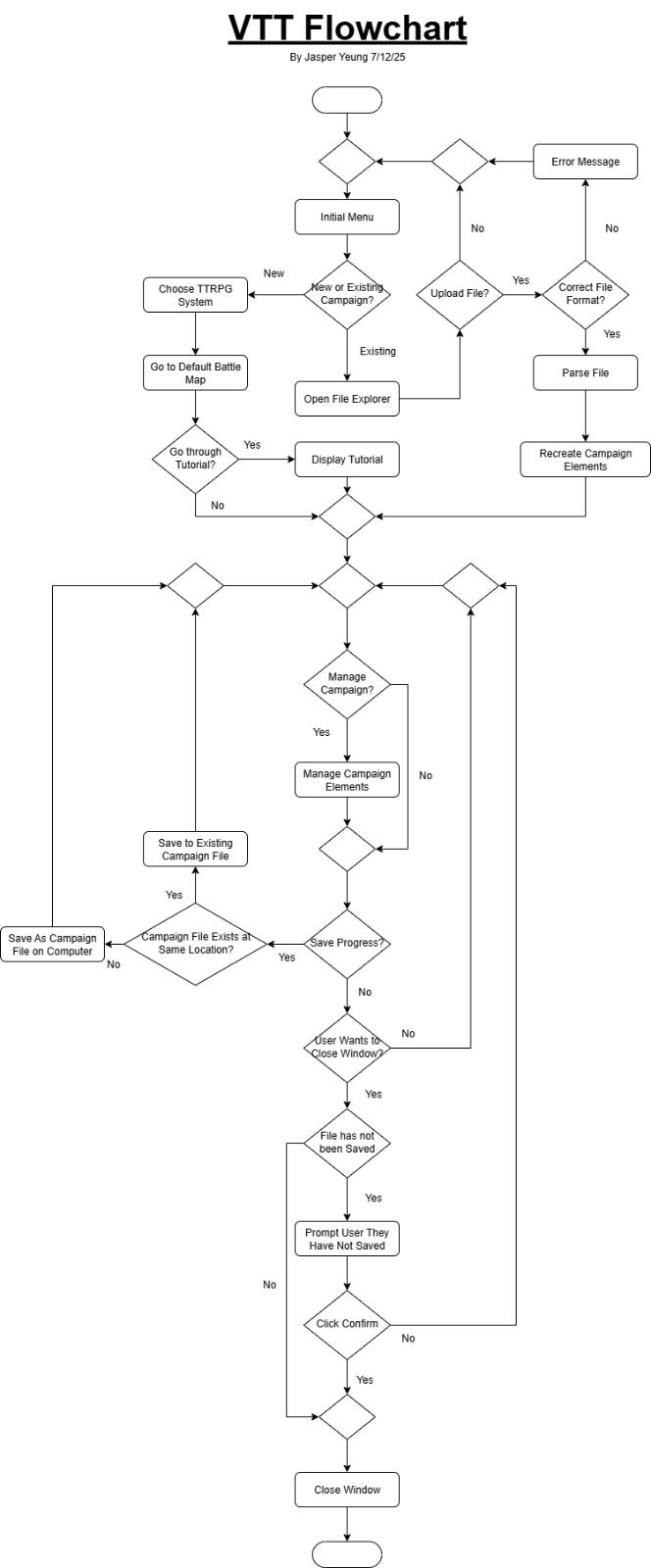


Figure : Flowchart

The Battle Map Mockup shown in *Figure 7*, provides an example of what the UI will look like for the Battle Map feature in both GM Mode and Streaming Mode. The buttons can be indicated as selected based on their color. Drop down menus are also present with their own fields and buttons. The main difference is that the Battle Map will have Tokens and other elements like shapes that are draggable across the map’s image. The sidebar menu has multiple tabs that will be expanded upon in the Sidebar Menu Mockup.

**Link:** https://github.com/Jyeung29/VTTWebApp/blob/main/Project\_Artifacts/Diagrams/Battle%20Map%20Mockup.jpg

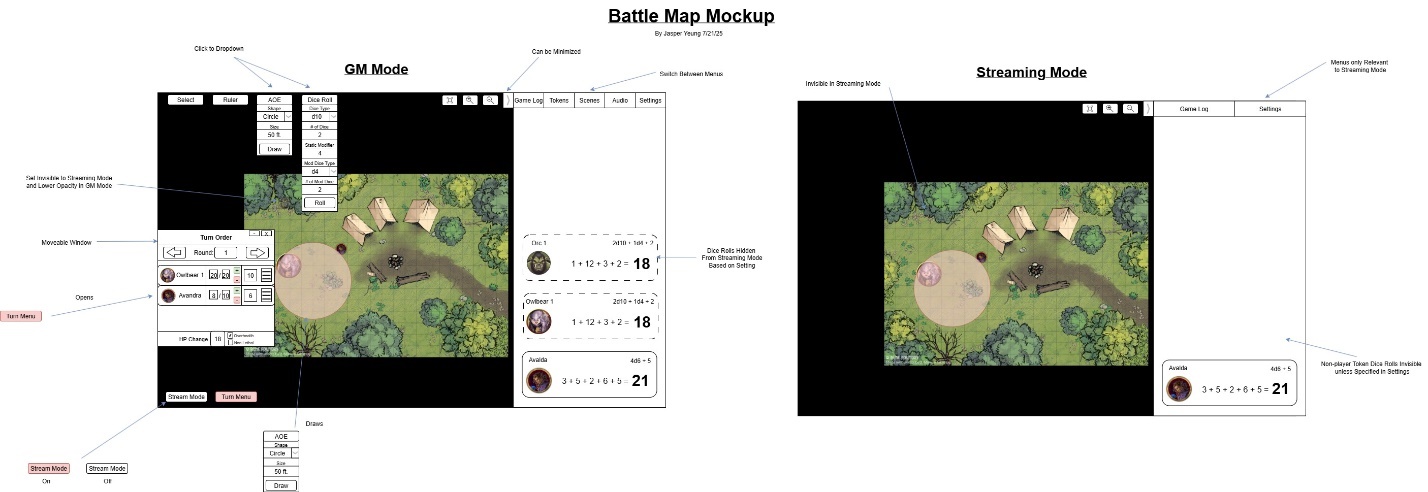


Figure : Battle Map Mockup

The Roleplaying Scene Mockup as shown in *Figure 8*, provides an example of what the Roleplaying Scene feature’s user interface will look like to the GM. The Roleplaying Scene still features draggable elements however, they are mainly full body character images. The toolbar still has the Select and Dice Roll buttons but an added effects button is added for visual effects. The Turn Menu and the Sidebar are the same as the Battle Map mockup.

**Link:** https://github.com/Jyeung29/VTTWebApp/blob/main/Project\_Artifacts/Diagrams/Roleplay%20Scene%20Mockup.jpg

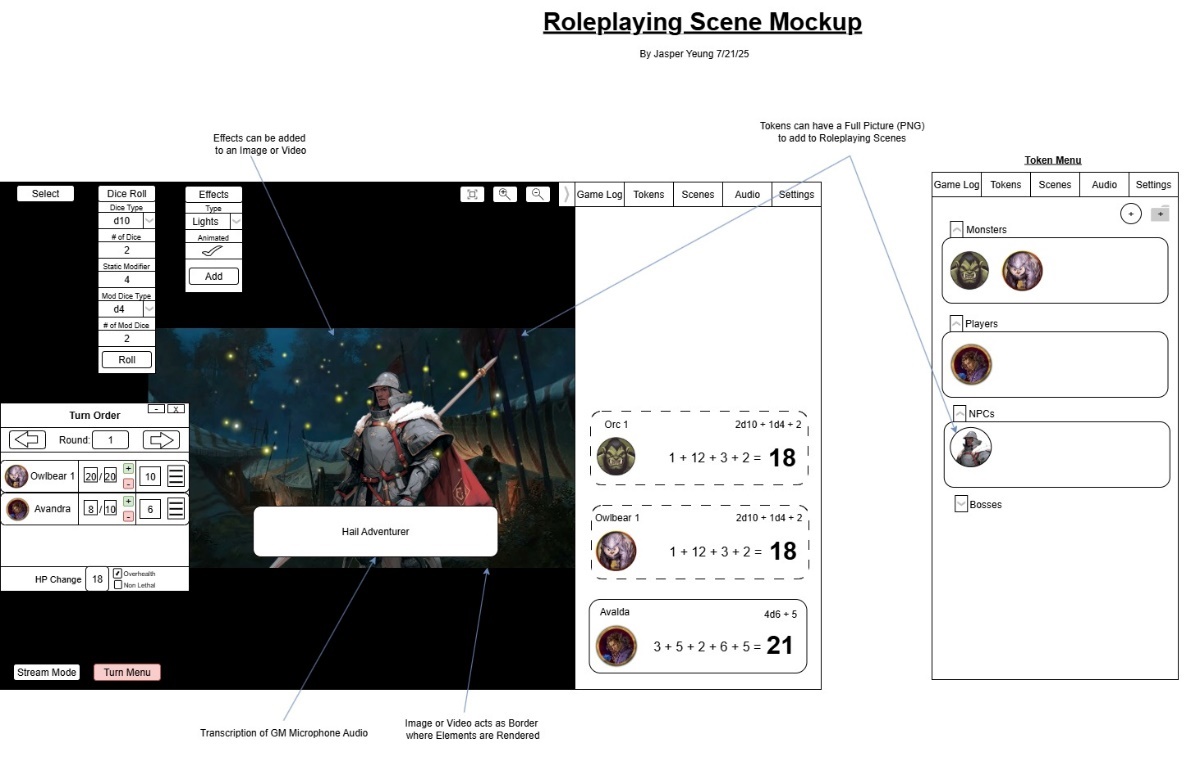


Figure : Roleplaying Scene Mockup

The Sidebar Menu Mockup in *Figure 9*, provide an example of what each tab of the Sidebar Menu will look like. The tabs include the Game Log, Audio Menu, Token Menu, and Scene Menu. The Game Log and Audio Menu are the only menus which the Streaming Mode are able to see. These menus work alongside the toolbar to provide GM’s with features to create immersive experiences for their players.

**Link:** https://github.com/Jyeung29/VTTWebApp/blob/main/Project\_Artifacts/Diagrams/Sidebar%20Menu%20Mockups.jpg

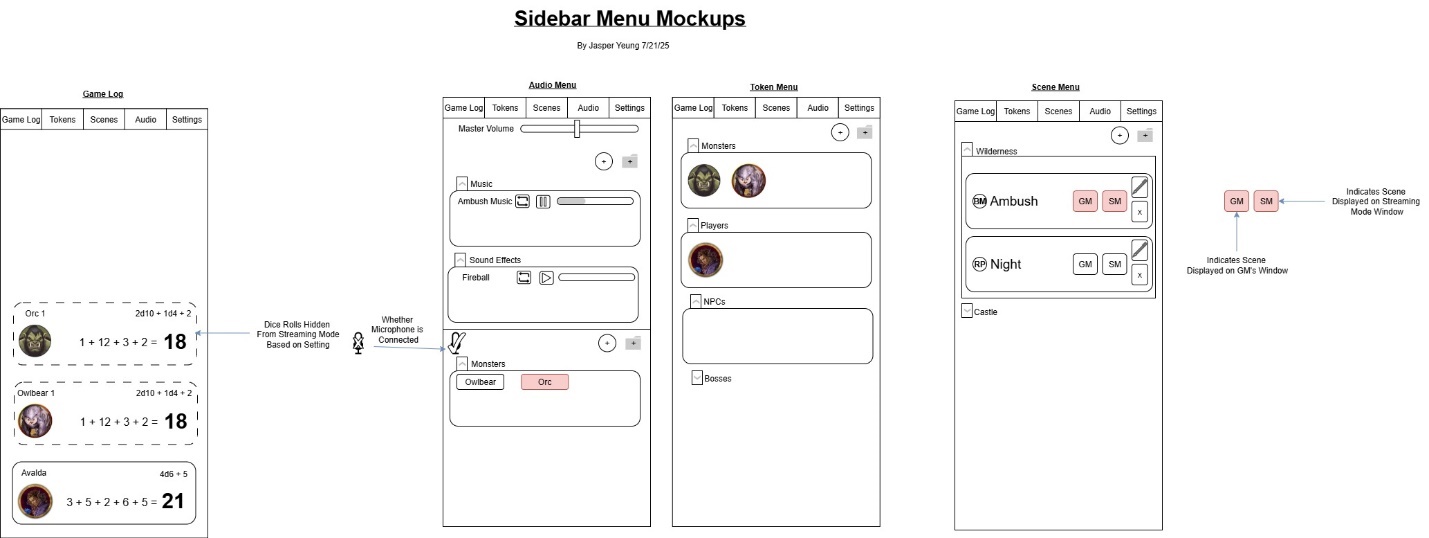


Figure : Sidebar Menu Mockup

Part 6 – Annotated Bibliography

TTRPG Sources

The sources in this section are all rule books of various TTRPG Systems. These sources were used to identify common objects and concepts which helped create the Domain Diagram and affected influenced the process of creating a software architecture that supports these TTRPG systems.

Bonner, Logan, et al. *Pathfinder Free Basic Rules*. 2nd Edition, Erik Mona, <https://roll20.net/compendium/pf2/Free%20Basic%20Rules#content>

Crawford, Jeremy, et al. “D&D Beyond Basic Rules.” *D&D Beyond*, <http://www.dndbeyond.com/sources/dnd/br-2024/1812-d-d-beyond-basic-rules>. Accessed 12 Aug. 2025

Peterson, Sandy, et al. *Call of Cthulhu Quick-Start Rules*. 7th Edition, Chaosium Inc., <https://www.chaosium.com/content/FreePDFs/CoC/CHA23131%20Call%20of%20Cthulhu%207th%20Edition%20Quick-Start%20Rules.pdf?srsltid=AfmBOoqj8HrXQEmPEWqR7tQDXyZP_-USc4q4wCJRXvP6TAxF42Y0rctk>

Tools and Methodologies Research

The sources in this section were used to inform the writing of the Design Specification document. These sources informed various sections including the tools used, testing plans, and metrics. Some sources were also used during the design phase in order to create professional diagrams.

Doglio, Fernando. “Top 7 Frontend Frameworks to Use in 2025: Pro Advice.” *Roadmap.Sh*, 2025, <https://roadmap.sh/frontend/frameworks>

Eseme, Solomon. “Top 5 TypeScript Frameworks (2022).” *Mastering Backend*, 1 Mar. 2021, <https://masteringbackend.com/posts/top-5-typescript-frameworks>

“JSON vs XML - Difference Between Data Representations - AWS.” *Amazon Web Services, Inc.*, <https://aws.amazon.com/compare/the-difference-between-json-xml/>. Accessed 12 Aug. 2025

Kirk, Aydan. “Building a Real-Time, Multi-User Collaborative Whiteboard Using Fabric.Js — Part I.” *Medium*, 24 Aug. 2020, <https://medium.com/@aydankirk92/building-a-real-time-multi-user-collaborative-whiteboard-using-fabric-js-part-i-23405823ee03>

“Not All Engineering KPIs Are Helfpul: How Great Teams Choose Better Metrics.” *Appfire*, <https://appfire.com/resources/blog/engineering-kpis-that-matter#cycle-time>. Accessed 12 Aug. 2025

“Software Testing Methodologies.” *Smartbear.Com*, <https://smartbear.com/learn/automated-testing/software-testing-methodologies/>. Accessed 12 Aug. 2025

“UML Association vs Aggregation vs Composition.” *Visual Paradigm*, <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-aggregation-vs-composition/>. Accessed 12 Aug. 2025

“Unit Testing in TypeScript.” *Refraction*, 14 Mar. 2024, <https://refraction.dev/blog/unit-testing-in-typescript>

Tool Documentation

The sources in this section are the documentation websites of the software development tools used in the project including libraries, frameworks, and languages. All citations but the Mdn Web Docs citation refer to the entire documentation websites due to numerous pages being referenced throughout the implementation of the project. The main information from these sources were classes, components, and demos each tool provided.

“Chakra UI.” *Chakra*, <https://chakra-ui.com/>. Accessed 12 Aug. 2025

“Documentation - Classes.” *TypeScript*, <https://www.typescriptlang.org/docs/handbook/2/classes.html>. Accessed 12 Aug. 2025

“Fabric.Js Javascript Library.” *Fabric.Js*, <https://fabricjs.com/>. Accessed 12 Aug. 2025

“Git Guide.” *GitHub*, 2025, <https://github.com/git-guides>

“Jest.” *Jest*, <https://jestjs.io/>. Accessed 12 Aug. 2025

“UI Events.” *Mdn Web Docs*, 28 Apr. 2025, <https://developer.mozilla.org/en-US/docs/Web/API/UI_Events>

Tutorials

TypeScript and React Setup

The sources in this subsection of tutorials were all used to set up TypeScript and React in the development environment. These sources gave information on what packages needed to be downloaded and how to set up a project in Visual Studio Code to begin implementation.

*React Tutorial Using TypeScript | Hands on React*. <https://handsonreact.com/docs/labs/react-tutorial-typescript>. Accessed 12 Aug. 2025

*TypeScript Tutorial*. <https://www.w3schools.com/typescript/index.php>. Accessed 12 Aug. 2025

“TypeScript Tutorial in Visual Studio Code.” *Visual Studio Code*, <https://code.visualstudio.com/docs/typescript/typescript-tutorial>. Accessed 12 Aug. 2025

“Using React in Visual Studio Code.” *Visual Studio Code*, 7 Aug. 2025, <https://code.visualstudio.com/docs/nodejs/reactjs-tutorial>

“Using TypeScript – React.” *React*, <https://react.dev/learn/typescript>. Accessed 12 Aug. 2025

TypeScript and React Basics

The sources in this subsection of tutorials were used to learn the basics of TypeScript and React. While some citations such as from W3 Schools provided comprehensive tutorials of basic TypeScript and React, other citations provided specific pieces of information that were needed during development. These pieces may have included TypeScript notations of functionality found in other programming languages or new functions that were unfamiliar.

DuMez, Kenny. “Operators in TypeScript.” *Graphite*, [https://graphite.dev/guides/typescript-operators. Accessed 12 Aug. 2025](https://graphite.dev/guides/typescript-operators.%20Accessed%2012%20Aug.%202025)

“Introduction to React.” *W3 Schools*, <https://www.w3schools.com/REACT/react_intro.asp>. Accessed 12 Aug. 2025

Mouskhelichvili, Tim. “How To Add Multiple Constructors In TypeScript?” *Tim Mousk*, 1 Nov. 2022, <https://timmousk.com/blog/typescript-multiple-constructors/>

Paudel, Nirjal. “Overloading TypeScript Constructors like C++ or Java with a Catch.” *Medium*, 18 Sept. 2023, <https://javascript.plainenglish.io/overloading-typescript-constructor-like-c-java-or-c-with-a-catch-b23267daec62>

Schulz, Marius. “Nullish Coalescing: The ?? Operator in TypeScript.” *Marius Schulz*, 6 Aug. 2020, <https://mariusschulz.com/blog/nullish-coalescing-the-operator-in-typescript>

“TypeScript Abstract Class.” *TutorialsTeacher*, <https://www.tutorialsteacher.com/typescript/abstract-class>. Accessed 12 Aug. 2025

“TypeScript ConstructorParameters<Type> Utility Type.” *GeeksforGeeks*, 14:55:27+00:00, <https://www.geeksforgeeks.org/typescript/typescript-constructorparameters-utility-type/>

“TypeScript Inheritance.” *GeeksforGeeks*, 23 July 2025, <https://www.geeksforgeeks.org/typescript/typescript-inheritance/>

“What Are ‘Implements’ Clauses in TypeScript ?” *GeeksforGeeks*, <https://www.geeksforgeeks.org/typescript/what-are-implements-clauses-in-typescript/>. Accessed 12 Aug. 2025

Feature Implementation

The sources in this subsection of tutorials were used to implement certain aspects of features in the project. These sources were used as inspiration and troubleshooting help while implementing the unique requirements necessary for feature functionality.

Aathi. “Answer to ‘How Do I Display Images from Google Drive on a Website?’” *Stack Overflow*, 28 Aug. 2018, <https://stackoverflow.com/a/52067077>

*How to Create Your Own Context Menu with Icons Using HTML, CSS & JavaScript*. Directed by dcode, 2024. *YouTube*, <https://www.youtube.com/watch?v=3jMykB29rOI>

klavsbuss. “Manually Set Bounding Box Size · Fabricjs/Fabric.Js · Discussion #7463.” *GitHub*, 29 Oct. 2021, <https://github.com/fabricjs/fabric.js/discussions/7463>

“Transformation Matrix – SVG Code – Aspose.SVG Guide.” *Aspose*, <https://docs.aspose.com/svg/net/drawing-basics/transformation-matrix/>. Accessed 12 Aug. 2025