Little Offices: Web Application

Daniel Cender

CST-451 Capstone Project Proposal

Grand Canyon University

Instructor: Professor Michael Landreth

Revision: 2

Date: April 9, 2023

# ABSTRACT

This project offers a web-accessible virtual meeting place, equipped with voice and text-based communication features. It seeks to close the gap between physical working spaces and the growing class of remote-first or hybrid-schedule workers in information technology roles. By eliminating a traditional list-view and “unread” feed of a traditional workplace chat application, task context-switching is reduced and the individual’s time is protected from inessential chatter or notifications.

This web application is designed in the style of a top-down traditional 2-dimensional RPG (role-playing game) to give a sense of movement and scale to the space where users meet. Users can authenticate with a 3rd-party OAuth source (Google), then select a digital avatar which will persist across whichever “worlds” they join. Once they’ve joined or created a World, they may enable voice chat or use the global message thread to send and receive text messages. If other users join their World, the telemetry of each user’s avatar is streamed through the server to all World members. This allows for near-real time browser rendering of where all the users are on the map. Using browser-based JavaScript APIs, users can hear voice chat content from other users whose avatars are within a limited tile range. A non-essential, but nice-to-have feature included in the project is a shared whiteboard feature, where all the users in a World can write and draw figures together.

| History and Signoff Sheet |
| --- |

**Change Record**

| **Date** | **Author** | **Revision Notes** |
| --- | --- | --- |
| April 2, 2023 | Daniel Cender | Initial draft for review/discussion |
| April 9, 2023 | Daniel Cender | Revisions including weekly project plan, designs, updated formatting. |
|  |  |  |

| **Overall Instructor Feedback/Comments** |
| --- |

| **Overall Instructor Feedback/Comments** |
| --- |

**Integrated Instructor Feedback into Project Documentation**

☐ Yes ☐ No

**Project Approval**

☐ Professor Mark Reha

# TABLE OF CONTENTS

[**ABSTRACT 2**](#_598cas1ovlf)

[**TABLE OF CONTENTS 4**](#_w85y9k71r1ta)

[**PROJECT OVERVIEW AND PROJECT OBJECTIVES 5**](#_1fob9te)

[**PROJECT SCOPE 8**](#_3znysh7)

[**PROJECT SUCCESS MEASURES 10**](#_2et92p0)

[**PROJECT HIGH-LEVEL SOLUTION 11**](#_tyjcwt)

[**PROJECT CONTROLS 14**](#_3dy6vkm)

# PROJECT OVERVIEW AND PROJECT OBJECTIVES

**State the Problem and Background**

This project is being coordinated as a proof-of-concept capstone to my course of study at Grand Canyon University. It serves the dual purpose of allowing me new avenues for expanding my skill with modern JavaScript frameworks to include more game-centric features like websocket networking.

Multiple target user groups are prime candidates for adopting a virtualized meeting place: WFH (Work from Home) professional personnel and geographically disconnected social groups.

For the past few years, particularly since the COVID-19 pandemic, many information workers have begun working a majority of their hours in a remote context. This new remote context comes with a new lack of presence with a distributed team, potential lack of clear communication, and possibly acute loneliness. These professionals could be served well by an application which allows them to recreate the communication and collaboration tools of a physical office, while maintaining the comfort and time-saving benefit of remote working conditions.

A smaller, but secondary group which might adopt are distributed social groups. Many people have friends and family who no longer live within a reasonable driving distance of each other. This application gives these groups an iteration on basic texting/calling methods for connecting with loved ones over the Internet.

**Christian Worldview**

For this project, the Christian worldview may apply to how communities are valued and prioritized, both for spiritual, emotional, and material goals. While the primary use-case might be for professionals, church groups could use this software for cases like we experienced during the COVID-19 lockdowns, when few people could meet as they ordinarily would. No software approximation can fully replace the reality of being present literally with other believers, but using CS skills to build tools like these can allow us to alleviate some loneliness and recreate some sense of togetherness during those times of isolation. This is merely a next-step in the long line of technologies used to create a sense of community, starting with the kind of written letters used in the first century.

If this project were part of a real-world startup, I’d expect the Christian worldview to guide me towards frugal development budgeting and spending. Many burgeoning startups take on more debt or investments than they truly need to begin providing a viable product, and it creates more loss and broken professional and personal relationships in the long run.

**Project Objectives**

1. Deliver an accessible user interface with authentication and preference selection features.
2. Deliver a basic, pleasing visual World map which can be traversed using keyboard navigation keys.
3. Deliver at least one interactable game item: a whiteboard backed by persistent storage.
4. Deliver a speedy server connection between multiple World users.
5. Restore connections to Worlds upon browser reloading and network failures.

**Challenges**

List the known challenges that will be used to measure project success.

* Developing the product with a lean budget. Ideally, the resulting cost will be as close to $0 as possible, and not exceeding $10 to host for the duration of the development roadmap and the project’s showcasing.
* Successfully harnessing WebRTC APIs for realtime audio transmissions. Many applications offering voice chat use some software processing to smooth audio data before sending it to the recipient. I may have to find examples and write some small audio processing utilities to achieve a decent transmission quality.
* Framerate: Utilizing visual assets and a rendering engine will require attention to the rendering update speed of the application. This needs to remain passable, at around 30 fps minimum at all times.

**Benefits and Opportunities**

Successful completion of this project will provide a base proof-of-concept for a fairly complex piece of collaboration software. It will allow me to, at the very minimum, showcase the tool in a personal portfolio site or on a resume. Beyond that, I can use lessons learned in its development to publish instructional web posts or video walkthroughs to benefit other learning developers and drive traffic to my professional profiles on LinkedIn or GitHub. Although it’s doubtful this tool would ever become a charged product for use on the open web, I will likely develop utility code pieces which will be useful in either future software roles or in personal web-based projects.

# PROJECT SCOPE

1. Give a clear, concise statement that states the scope of the project. This should also include items that are to be out of scope.
   1. This project connects to Google identity providers to allow OAuth authentication.
   2. User preferences, like a username and selected avatar, are stored in a persistent database.
   3. Generated Worlds and their joined users will be saved to a persistent database.
   4. Users will be able to send/receive messages in real time while in a World.
   5. Users will be able to send voice chat data and hear other active users while in a World together.
   6. Users will not be able to share webcam data, although that makes use of similar browser functions as voice chat.
   7. Users will be able to collaborate on a shared HTML Canvas whiteboard tied to each World.
   8. Users will be able to delete the Worlds they own.
   9. Project will not guarantee the ability to delete one’s user
   10. Project will not guarantee the ability to mute/unmute their microphone while in a World.
   11. Project will not provide the ability to create a custom avatar; all avatar options will be whole pre-selected sprite assets.
2. Use the template to list all known stakeholders and contacts, if applicable, including self (for some projects self may be the only name listed)

| Stakeholder Name | Role(s) | Responsibilities |
| --- | --- | --- |
| Self | Designer, Developer, Project Manager | Defining project requirements and roadmap, designing visual features, architecting software deployment structures, and developing full-stack software implementation. |

1. List the work breakdown required to satisfy the project objectives. Identify teams and other resources that may be required to successfully complete the project.
   1. Work task br

| Work Breakdown Structure | | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Task | Dependencies | Status | Effort Hours | Cost | Start Date | Planned Completion | Estimate to Completion | Actual Completion | Resource |
| 1 | Select Infrastructure/Tooling |  | Incomplete | 4 hours | 0 | 4/2/2023 | 4/3/2023 |  |  | Time |
| 2 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

# PROJECT SUCCESS MEASURES

The success of this project will be determined by the ability to implement the desired features in the allotted timeline. Some minor constraints will be enforced around site security and performance. However, the performance of the site is expected to be suboptimal with the fast timeframe given for its entire development. With more time, and allotting more money to provision faster cloud resources and multiple availability zones, the performance could be improved over time.

| Project Completion Criteria |
| --- |
| 1 - Project is developed within planned timeframe. |
| 2 - Project is developed within budget. |
| 3 - Project functions as expected in the required feature set. |
| 4 - Project functions within an expected latency limit (under 100ms responses from server to multiple clients). |
| 5 - Project is secured using a 3rd party OAuth method (ideally Google sign-in). |
| 6 - Project is free of obvious or intrusive bugs/undesired behavior. |

Use the template to list the project assumptions and constraints, if applicable. An assumption is an educated guess that a likely condition or circumstance is presumed to be true. A constraint is a limiting condition or circumstance that defines the project boundaries. Assumptions allow the project to succeed. Constraints restrict or limit the project execution.

| Assumptions and Constraints | | | | | |
| --- | --- | --- | --- | --- | --- |
| ID | Description | Comments | Type | Status | Date Entered |
| 1 | A basic Node.js server can be hosted for free on Google Cloud Provider. |  | Assumption | Unconfirmed | 4/2/2023 |
| 2 | Free or cheap assets are available which will suit the needs of the product to build a visual World map and avatar selector. |  | Criteria | Unconfirmed | 4/2/2023 |

# PROJECT HIGH-LEVEL SOLUTION

**Introduction**

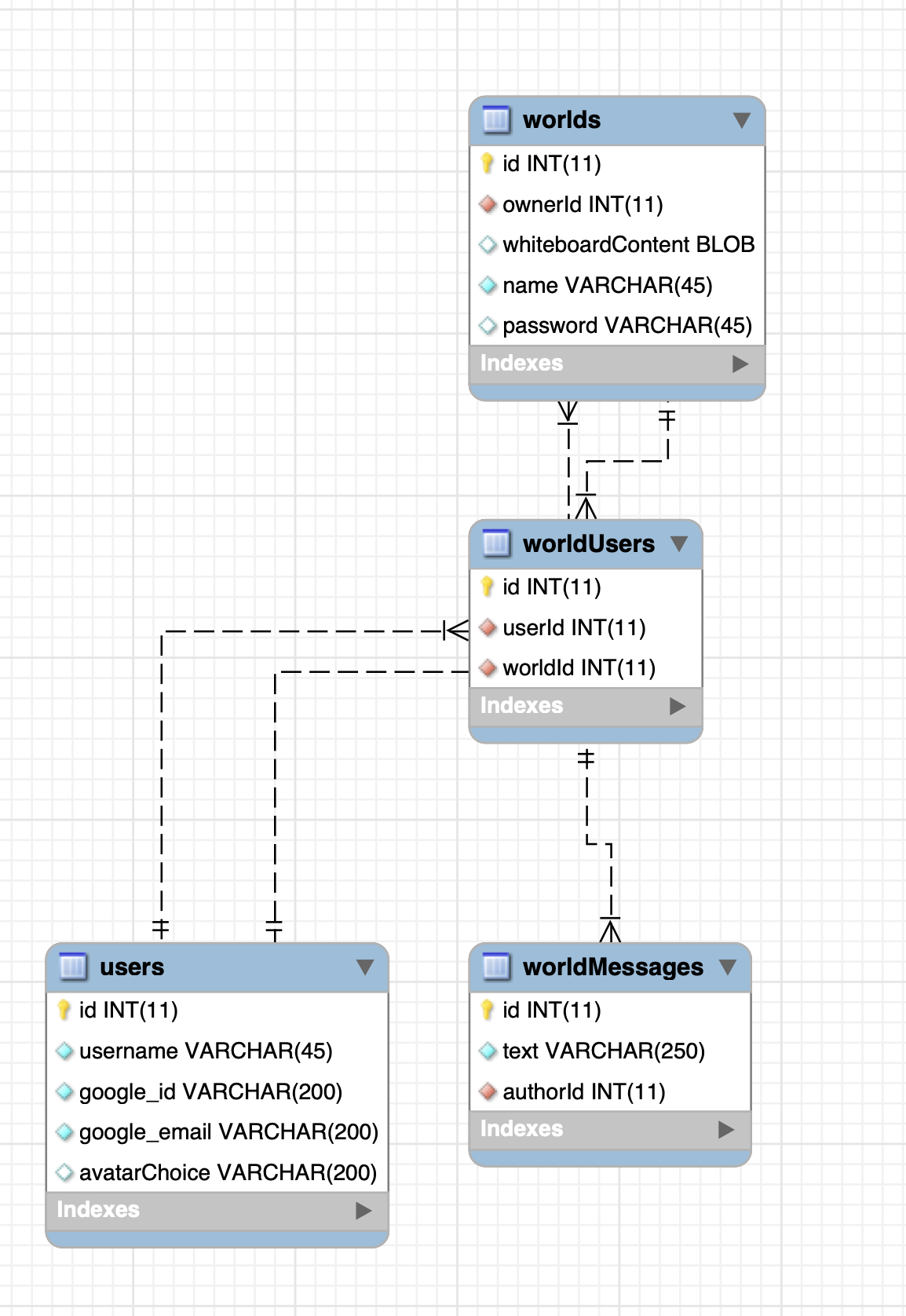
The main objective in this project is to develop a web-first site outfitted with an array of communication tools. Ideally, this will include an audible method, and at least one visual method. In addition to producing a useful product, it will test my abilities to combine multiple web technologies into a single application, while maintaining reasonable, fast performance.

I’ve assumed the functionality of voice chatting and sharing an editable drawing surface from within the browser. Due to my prior experience in JavaScript development, I have full confidence the WebRTC and JavaScript Canvas programming APIs (built into the JavaScript engine in every modern browser) can handle these pieces of functionality, with some extra programming to provide smoother user experiences.

**Solution**

This application combines a mixture of stateful and stateless operations, which makes it an interesting and slightly tricky project to finish. Information from the World whiteboard, user avatar movement, and messages will be streamed to all connected World users synchronously. The data surrounding who is connected to which Worlds, which avatar each user has selected in their Settings, and their authenticated Google account ID are all going to need to persist even if the server instance providing this functionality crashes.

At this stage of development, I’ve identified the need for a relational database system. I’ve chosen MySQL due to my prior experience with the tool, although SQL Server or PostgreSQL would work just as well for this project. The ER (Entity Relationship) Diagram below is a database schema I believe will satisfy the requirements for this short project.



In addition to a stateful data storage solution, I’ll need to host a web-available server which can pipe client data packets from one client to all the connected World users. A great solution is to use an open-source tool or framework for managing these connections. It will cut down on development time and allow for more thorough testing. Two possible frameworks which could be used to write the application server are:

* Colyseus (*Multiplayer server: Colyseus: Simple & fast multiplayer game creation*, n.d.)
* Socket.io (*Socket.io,* n.d.)

Both provide powerful tooling for establishing connections between the server and browser client for specific “rooms” (or Worlds, in this use-case). More research is required to determine which will suit the product best.

For the user interface, I will need to make use of self-created or freely available visual assets, along with a rendering toolset. *Itch.io* is a platform with many free-to-use art assets which could compose the main World visuals and provide avatar options for users (*Top game assets*, n.d.). These assets will be useless without loading them into an engine which I can manipulate to create a core “game loop” for the application. I’ll be using *React Three Fiber* as the rendering engine behind the main World-based application functionality (*React-Three-Fiber: Introduction*, n.d.). This tool offers powerful texture and image loaders, along with some light physics engine capabilities.

# PROJECT CONTROLS

| Risk Management | | | | |
| --- | --- | --- | --- | --- |
|  | **Risk Probability** | **Risk Impact** |  |  |
| **Event Risk** | **(high, medium, low)** | **Risk Mitigation** | **Contingency Plan** |
| Over Budget | Low | Will decommission or downgrade cloud resources, causing significant app downtime. | Research cloud tools with extensive free tiers and low consumption pricing.  Opt for free lower-speed tiers when possible. | Set billing account alerts when any cloud charges are accumulated.  Downgrade production-grade cloud resources to development versions. |
| Over Deadline | High | Will result in publishing an incomplete, insecure, or half-designed solution for the product case. | Re-evaluate project milestones and progress weekly. Adjust scope and remove excessive features as necessary.  Utilize “feature flags” for different parts of the UI and functionality. | Stub out features with “coming soon” placeholders.  Use feature flags to turn off functionality if incomplete at the time of launch. |
| Impossible Scope | Low | Will require cutting essential features or paying for 3rd-party implementations, testing budget limits. | Find open-source and well-documented tools to eliminate reinventing established patterns. | Update promised deliverables to consumers/stakeholders.  Publish a roadmap with original scope items on an updated timeline, after initial launch.  Perform an expose on difficulties with launching as expected. Might possibly avoid some kickback from unsatisfied stakeholders. |

| Issues Log | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Project Impact** | **Action Plan/Resolution** | **Owner** | **Importance** | **Date Entered** | **Date to Review** | **Date Resolved** |
| 1 |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |

| Change Control Log | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Change Description** | **Priority** | **Originator** | **Date Entered** | **Date Assigned** | **Evaluator** | **Status** | **Date of Decision** | **Included in Rev. #** |
| 1 |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |

1. Use the template to describe how the end user is involved in the software development, if applicable. Include relevant information about meetings, reviews, presentations, etc.

* No user or outside stakeholder involvement yet in the development process.

| Roles and Responsibilities | | | |
| --- | --- | --- | --- |
| Name | Team | Project Role | Responsibility |
|  |  |  |  |
|  |  |  |  |

**Project Cost and Schedule**

1. Create a spreadsheet of costs related to the scope of the project, with all necessary material and elements required to accomplish it effectively, and the allocated resources. Note: If the project being designed will not require any cost calculations, please state that here.
   1. Currently, there are no expected costs to be calculated. All product deliverables should be achieved through a set of free-tier cloud resources. If the requirements fall beyond the limits of a free tier, an updated project plan document will be revised to include a cost breakdown.
2. Create a project schedule after all project tasks have been defined and prioritized.
   1. The project schedule will be finalized and present in the following week.
3. Set a programming schedule by implementing work breakdown and task time estimates. Create a timeline with dates for completion of key components of the project.
   1. The programming task schedule will be implemented in the following week. A rough outline of the expected development progress has been provided below.

**By Week**

* Week 1: Concept selected, tool research started, initial project proposal completed.
* Week 2: All screens fully wireframes and final tool selection completed. Logical system diagrams are complete. User stories are complete.
* Week 4: Higher-fidelity designs complete. Test cases and acceptance criteria for all user stories written up. Persistent Data Storage all wired up and functional. Can save Google Auth, Username, and sprite of choice.
* Week 5: Allow multiple server user connections. Send/receive data between clients over NestTS server.
* Week 6: Graffiti whiteboard functionality, send/receive data from all clients to keep board updated at all times.
* Week 7: Proximity Voice chat. Includes selecting proximity radius when creating server channel.
* Week 8: Final fixes, polish, notes on future roadmap, total budget/time investment for the project.

**Appendix A – References**

Google. (n.d.). *Overview | authentication | google developers*. Google. Retrieved April 2, 2023, from https://developers.google.com/identity/gsi/web/guides/overview

*Multiplayer server: Colyseus: Simple & fast multiplayer game creation*. Colyseus. (n.d.). Retrieved April 2, 2023, from <https://www.colyseus.io/>

*React-Three-Fiber: Introduction*. React Three Fiber Documentation. (n.d.). Retrieved April 2, 2023, from <https://docs.pmnd.rs/react-three-fiber/getting-started/introduction>

*Socket.io*. SocketIO RSS. (n.d.). Retrieved April 2, 2023, from <https://socket.io/>

*Top game assets*. itch.io. (n.d.). Retrieved April 2, 2023, from <https://itch.io/game-assets>

webrtcHacks. (2023) adapter [Source code]. <https://github.com/webrtcHacks/adapter>

**Appendix B – Copyright Compliance**

The following potential/utilitized tools are distributed under the MIT License, which allows for free redistribution and use, even for commercial products:

* Colyseus, license: <https://github.com/colyseus/colyseus/blob/master/LICENSE>
* Socket.io, license: <https://github.com/socketio/socket.io/blob/main/LICENSE>
* React-three-fiber, license: <https://github.com/pmndrs/react-three-fiber/blob/master/LICENSE>

No specific art assets have been selected for use in the project, but all licenses or links to artist disclosures will be provided in this section in future project plan updates.

The following tools are distributed under the BSD 3-Clause "New" or "Revised" License:

* adapter, license: <https://github.com/webrtcHacks/adapter/blob/main/LICENSE.md>