Capstone Project: Little Offices

Author(s): Daniel Cender

CST-451 Capstone Project Requirements Document

Grand Canyon University

Instructor: Professor Michael Landreth

Revision: 1

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 9, 2023 | Daniel Cender | Initial draft for review/discussion |
|  |  |  |
|  |  |  |

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

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

**Integrated Instructor Feedback into Project Documentation**

☐ Yes ☐ No

**TABLE OF CONTENTS**

[**ABSTRACT 2**](#_92s8idbxp22j)

[**FUNCTIONAL REQUIREMENTS 5**](#_30j0zll)

[Use Cases 5](#_erqerz7wgn71)

[**NON-FUNCTIONAL REQUIREMENTS 6**](#_1fob9te)

[Use Cases 6](#_t6g14co2zd2b)

[**TECHNICAL REQUIREMENTS 7**](#_3znysh7)

[Code Utilities 7](#_5l2x4499xjiq)

[Code Hosting 7](#_9976z9ssi0hk)

[Data Persistence 7](#_8u670z7f9dwg)

[Security 7](#_v6bxb8amgv36)

[Application Deployment 7](#_2d5j1uq68v03)

[**LOGICAL SYSTEM DESIGN 9**](#_tyjcwt)

[World Creation/Joining Logical Diagram 9](#_qc7smmqudtdg)

[Server Chat System Logical Diagram 10](#_nh9uhxn4wwhw)

[Voice Chat System Logical Diagram 11](#_ajyrn64spt8s)

[Server Whiteboard Edit System Logical Diagram 12](#_4b52q4nnu3pm)

[**USER INTERFACE DESIGN 13**](#_3dy6vkm)

[Site Map 13](#_3xme0phizbe)

[Page Wireframes 14](#_xcedguyh7pgi)

[**SECURITY 21**](#_12qbevqok51v)

[User Roles 21](#_960yfjy5gepd)

[**REPORTS DESIGN 23**](#_1t3h5sf)

# FUNCTIONAL REQUIREMENTS

## Use Cases

All use case stories are captured in a separate, attached Excel document: CST-451FunctionalRequirementStories.xlsx

All out-of-scope functional use case stories will be documented in the table below:

| **Use Case or User Story** | **Approval Date** | **Justification** |
| --- | --- | --- |
|  |  |  |

# NON-FUNCTIONAL REQUIREMENTS

## Use Cases

All non-functional use case stories are captured in a separate, attached Excel document: CST-451NonfunctionalRequirementStories.xlsx

All out-of-scope functional use case stories will be documented in the table below:

| **Use Case or User Story** | **Approval Date** | **Justification** |
| --- | --- | --- |
|  |  |  |

# TECHNICAL REQUIREMENTS

All the notable tools and technologies used in the project.

## Code Utilities

* React-three-fiber: A frontend framework which provides the classes available in the rendering framework ThreeJS as a set of ReactJS components. <https://docs.pmnd.rs/react-three-fiber/getting-started/introduction>
* Colyseus: A barebones NodeJS multiplayer game server framework, provides helpful tools for facilitating state updates between multiple game clients. <https://docs.colyseus.io/colyseus/>
* ReactJS: A powerful web development framework for building modern, state-driven websites and apps. <https://react.dev/>
* adapter: A compatibility “shim” utility to smooth over inconsistencies between browser WebRTC API implementations. <https://github.com/webrtcHacks/adapter>
* PeerJS: Simple WebRTC wrapper for easy browser-to-browser audio/video calling. <https://github.com/peers/peerjs>

## Code Hosting

* GitHub: Code hosting and CI/CD pipeline tools. <https://github.com/>

## Data Persistence

* MySQL Server: A highly-scalable database server. <https://www.mysql.com/>

## Security

* Google Identity: A secure method for supporting OAuth 2.0 user identification with ease. <https://developers.google.com/identity/gsi/web/guides/overview>

## Application Deployment

The application will be deployed using a tool from Google Cloud called App Engine. App Engine has simple connection bindings to popular code repository platforms, so wiring up a continuous integration and continuous delivery (CI/CD) pipeline should be trivial. App Engine can be read about here: <https://cloud.google.com/appengine/docs/standard/nodejs/building-app/deploying-web-service>

NOTE: Once the technical requirements have been completed, there may be situations where technologies or tools may need to be taken out of scope or changed, possibly due to technical challenges or timeline challenges. Any technologies or tools that are taken out of scope or changed once the project development has started must be approved by the mentor and instructor with justification as to why the functionality is being removed from the project. The following must be updated if any technologies or tools are taken out of scope or changed:

| **Technology or Tool** | **Approval Date** | **Justification** |
| --- | --- | --- |
|  |  |  |

# LOGICAL SYSTEM DESIGN

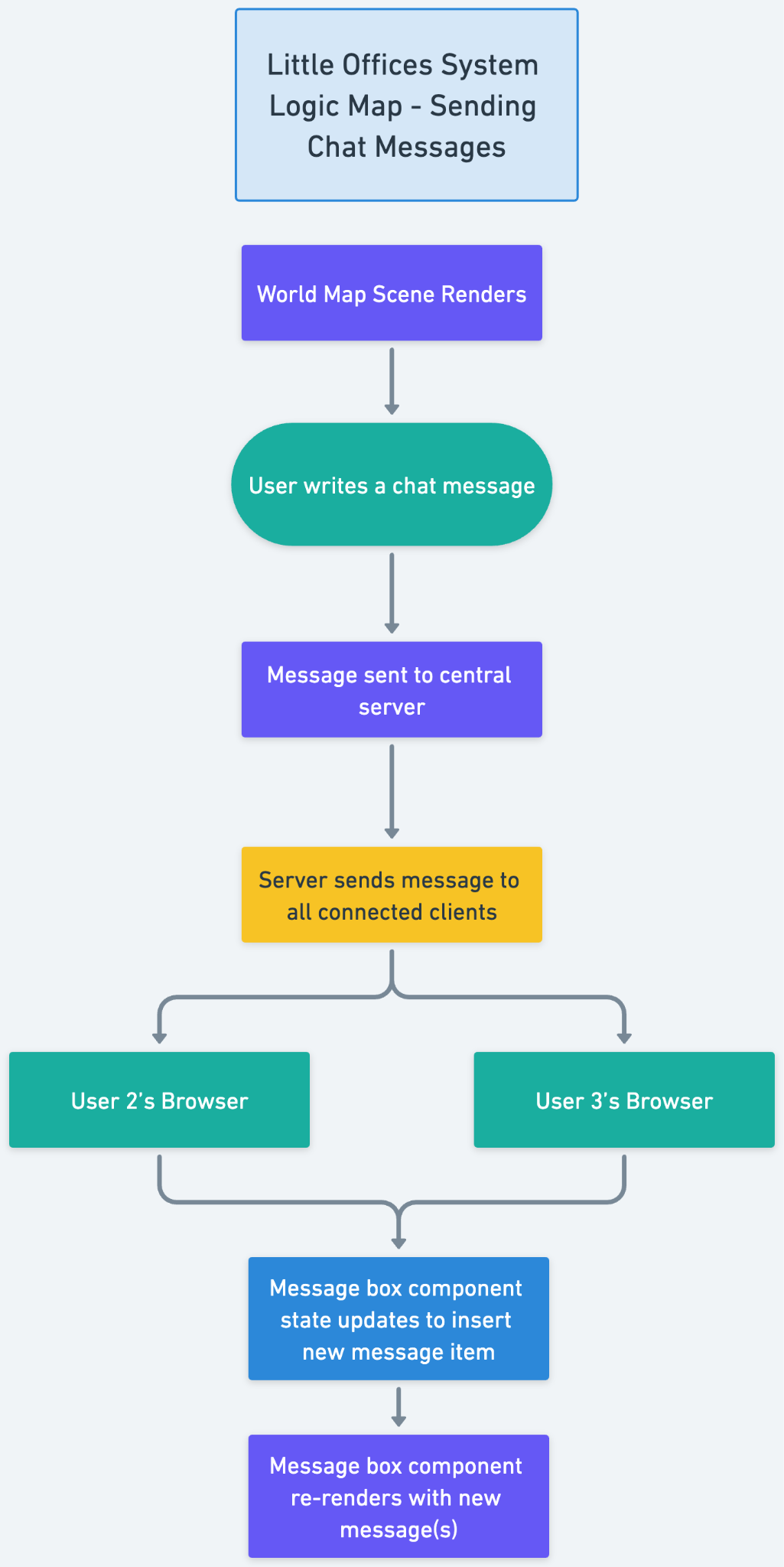
## World Creation/Joining Logical Diagram

## 

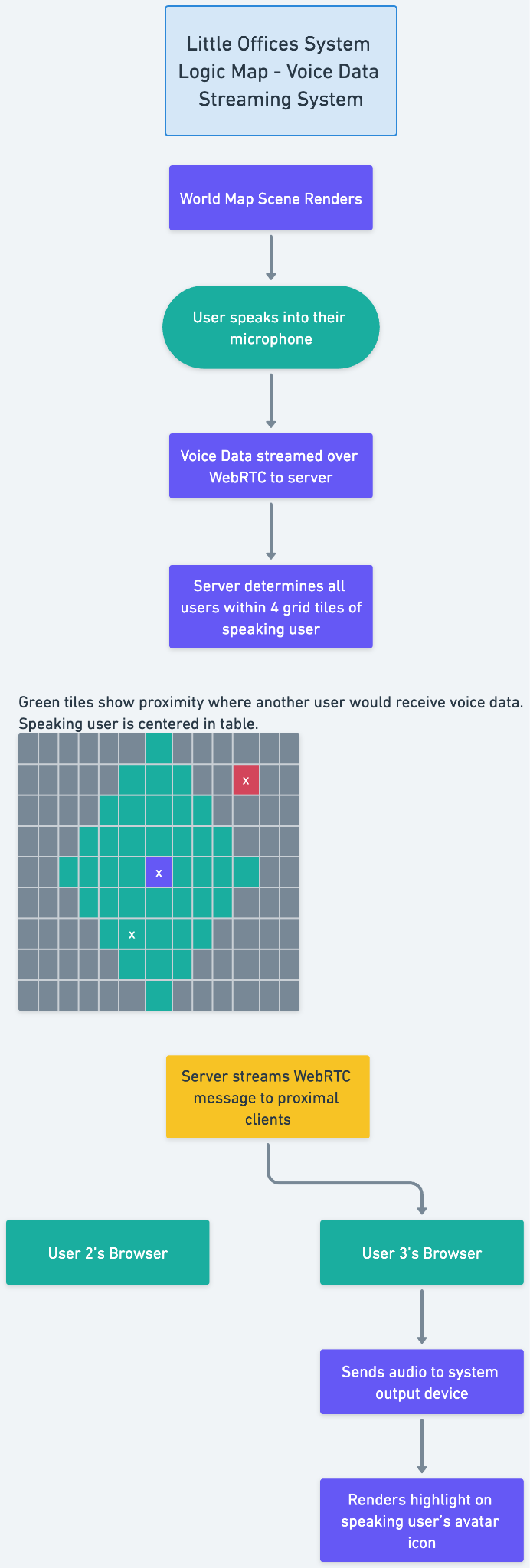
## 

## 

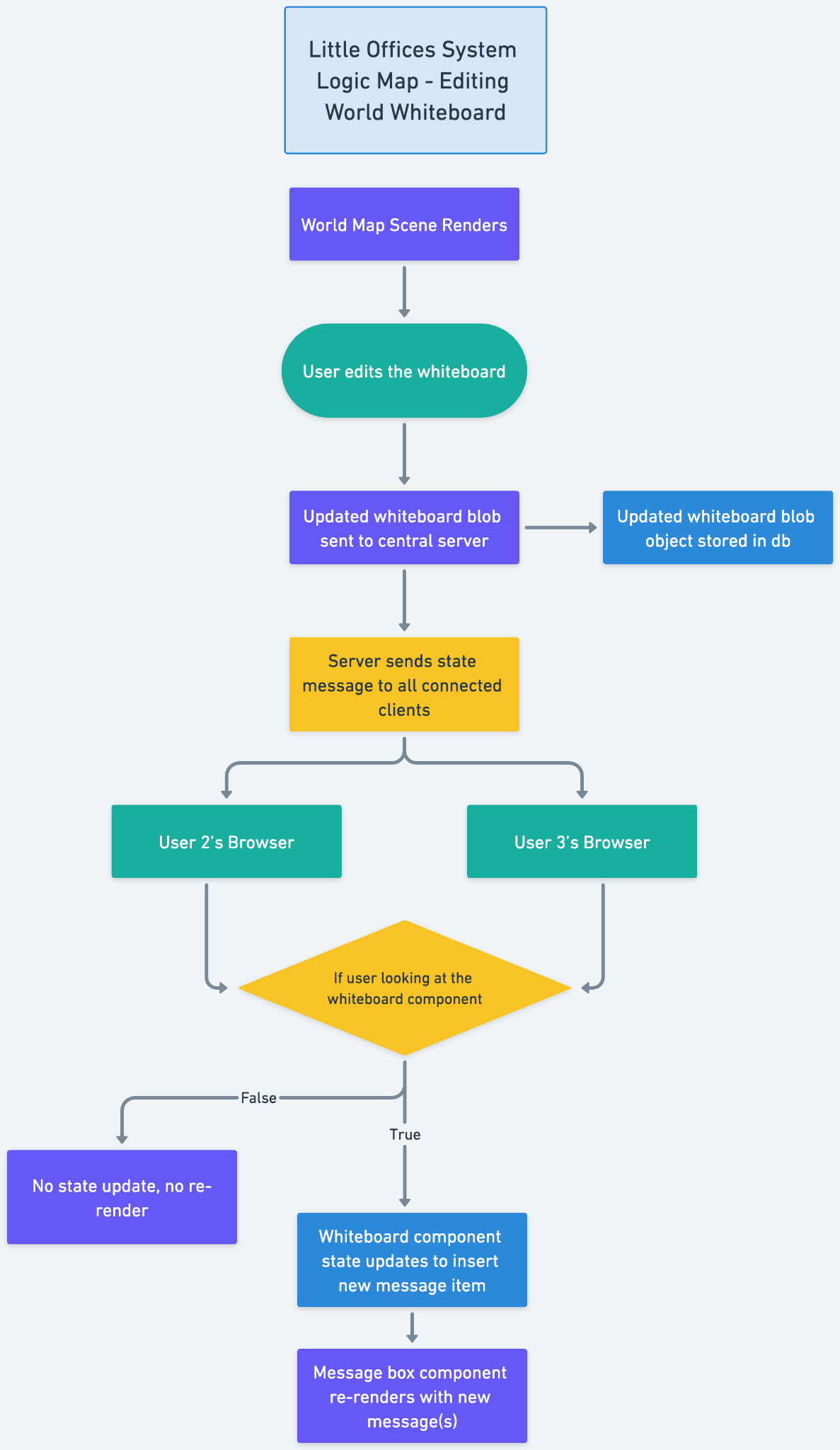
## Server Chat System Logical Diagram



## Voice Chat System Logical Diagram

****

## Server Whiteboard Edit System Logical Diagram

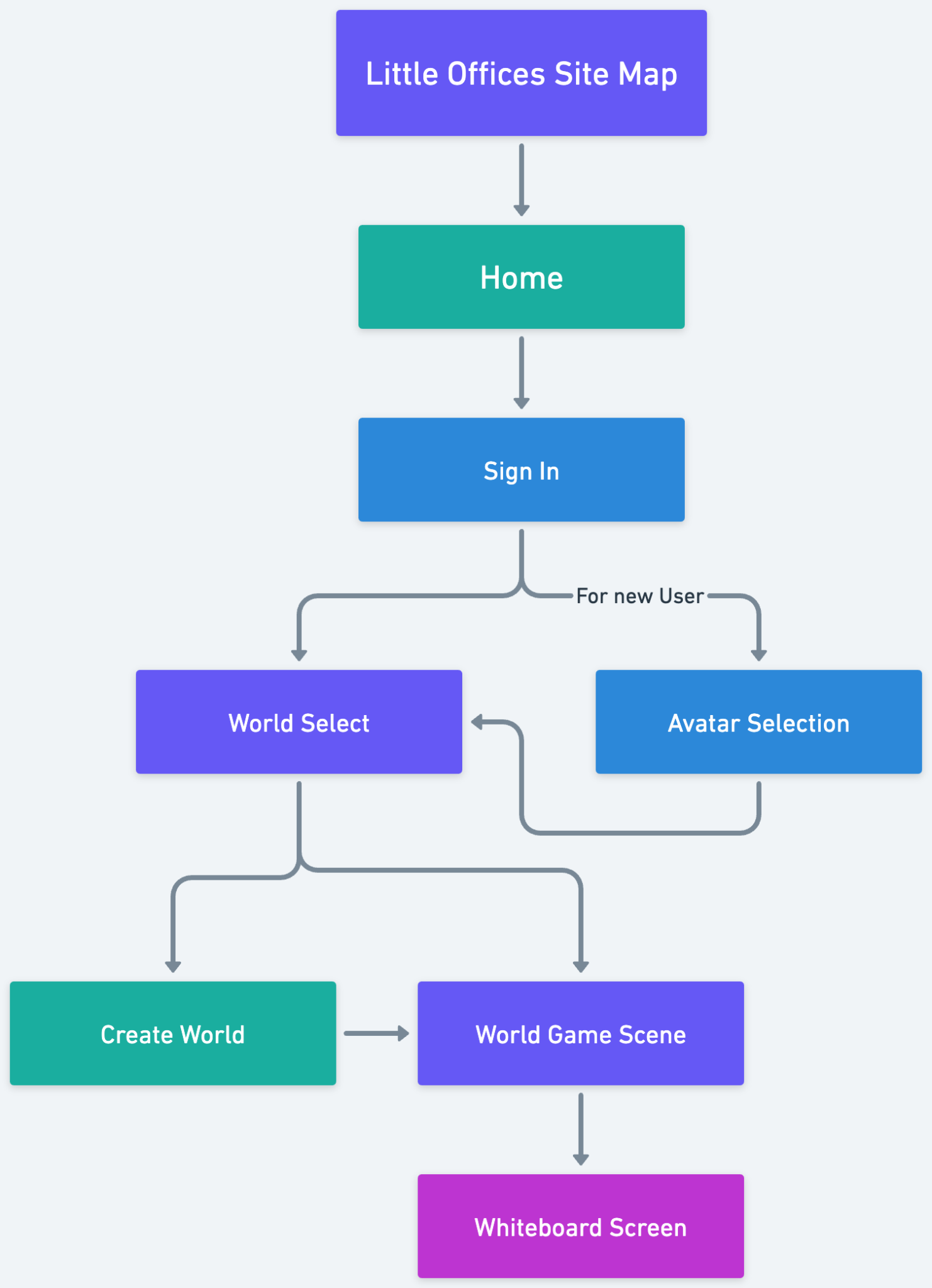
****

# USER INTERFACE DESIGN

A sitemap and user interface design diagram for each user interface screen in the application.

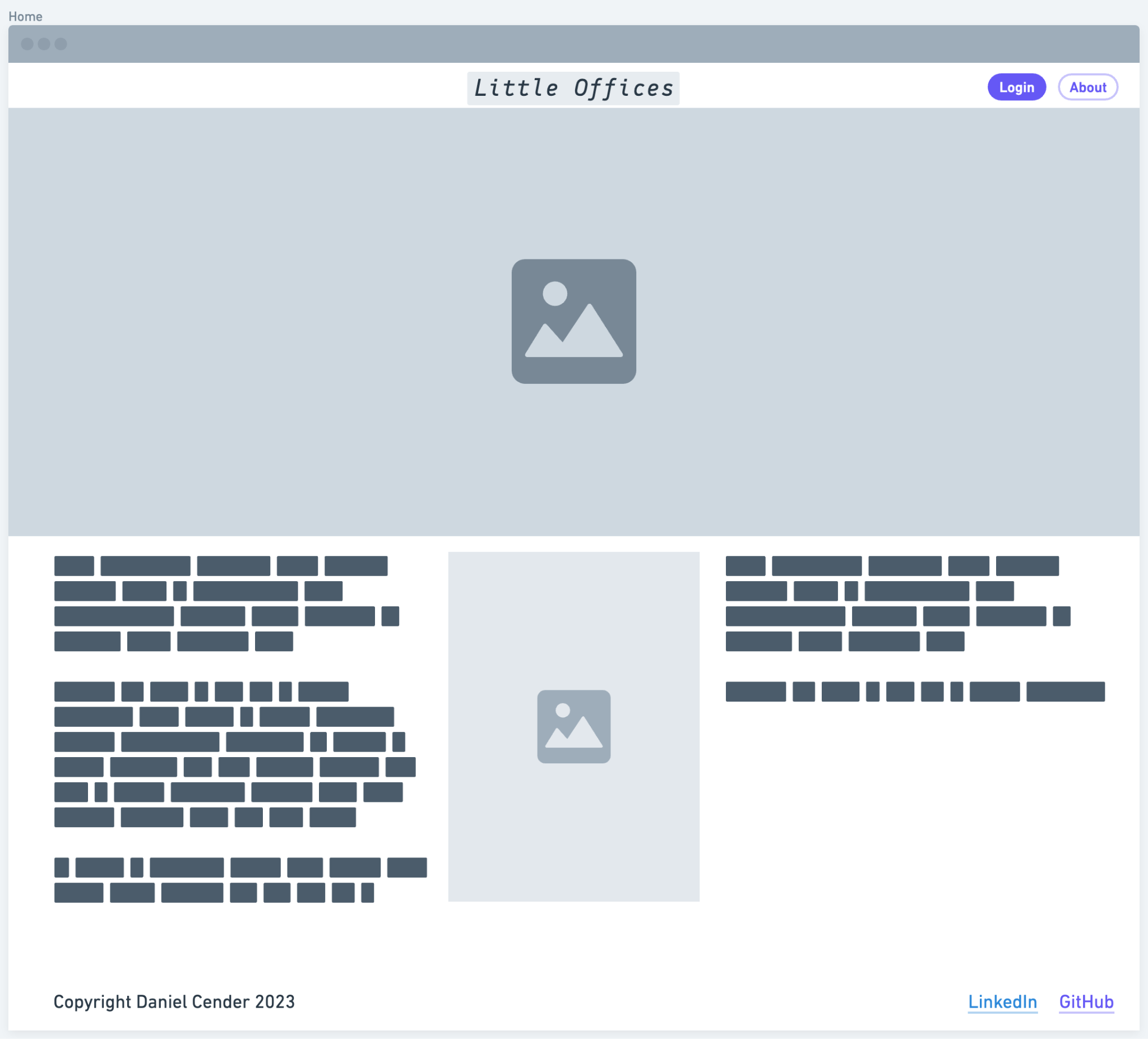
## Site Map

I’ve included a basic site map diagram, which only includes the individual pages a user may navigate through, and how they connect to each other, culminating in the rendered World game scene.

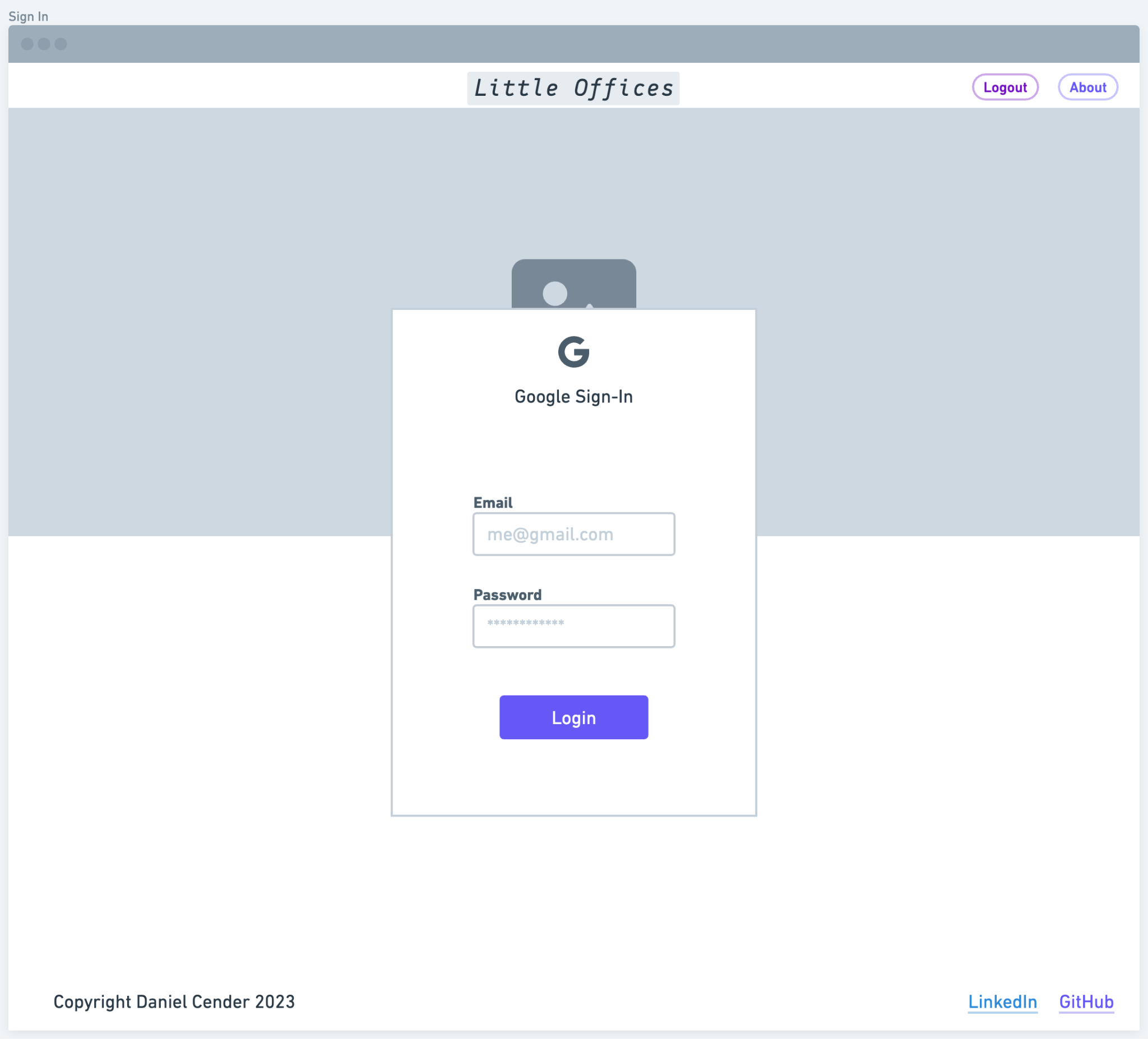


## Page Wireframes

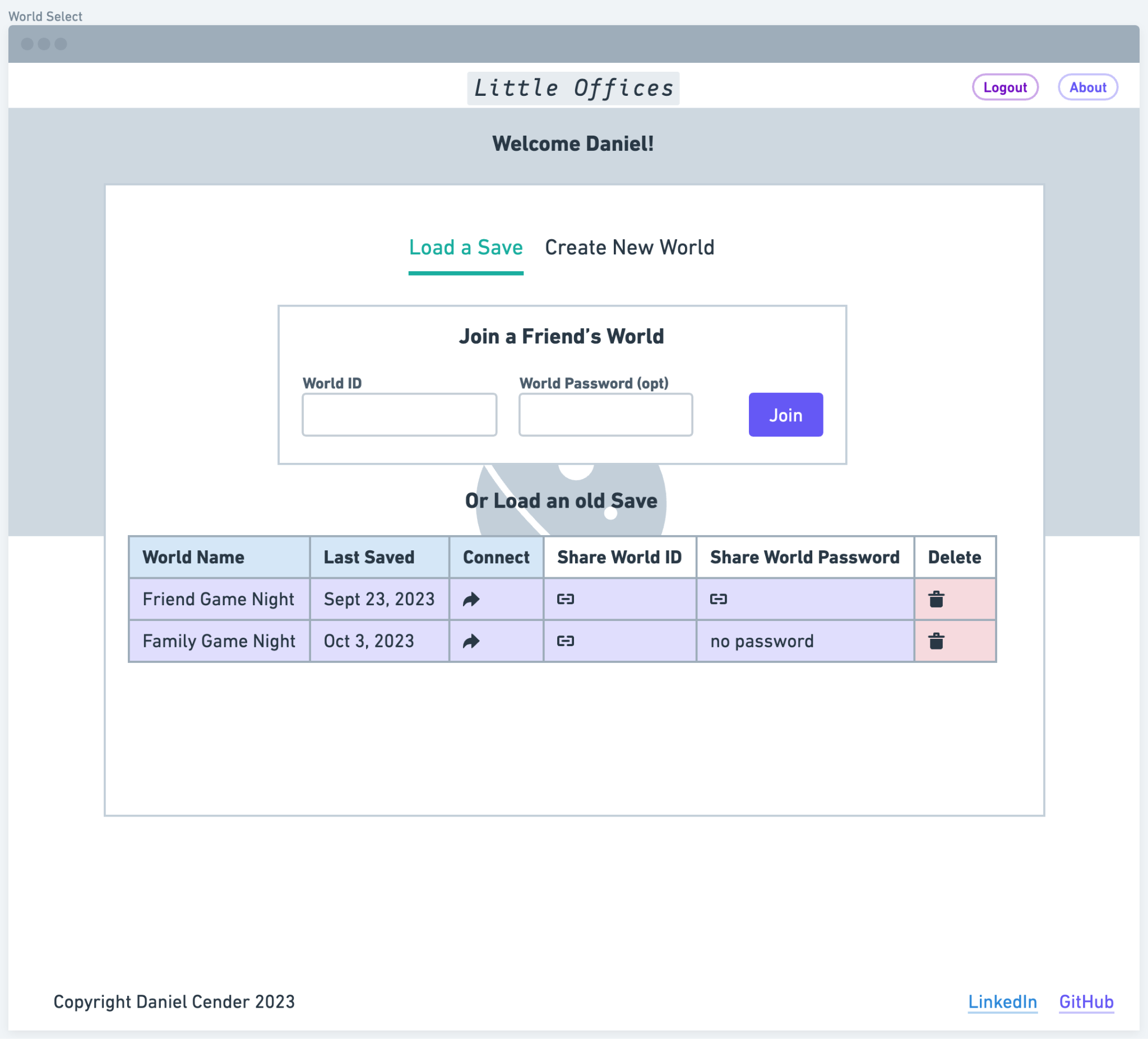
### *Home page:*



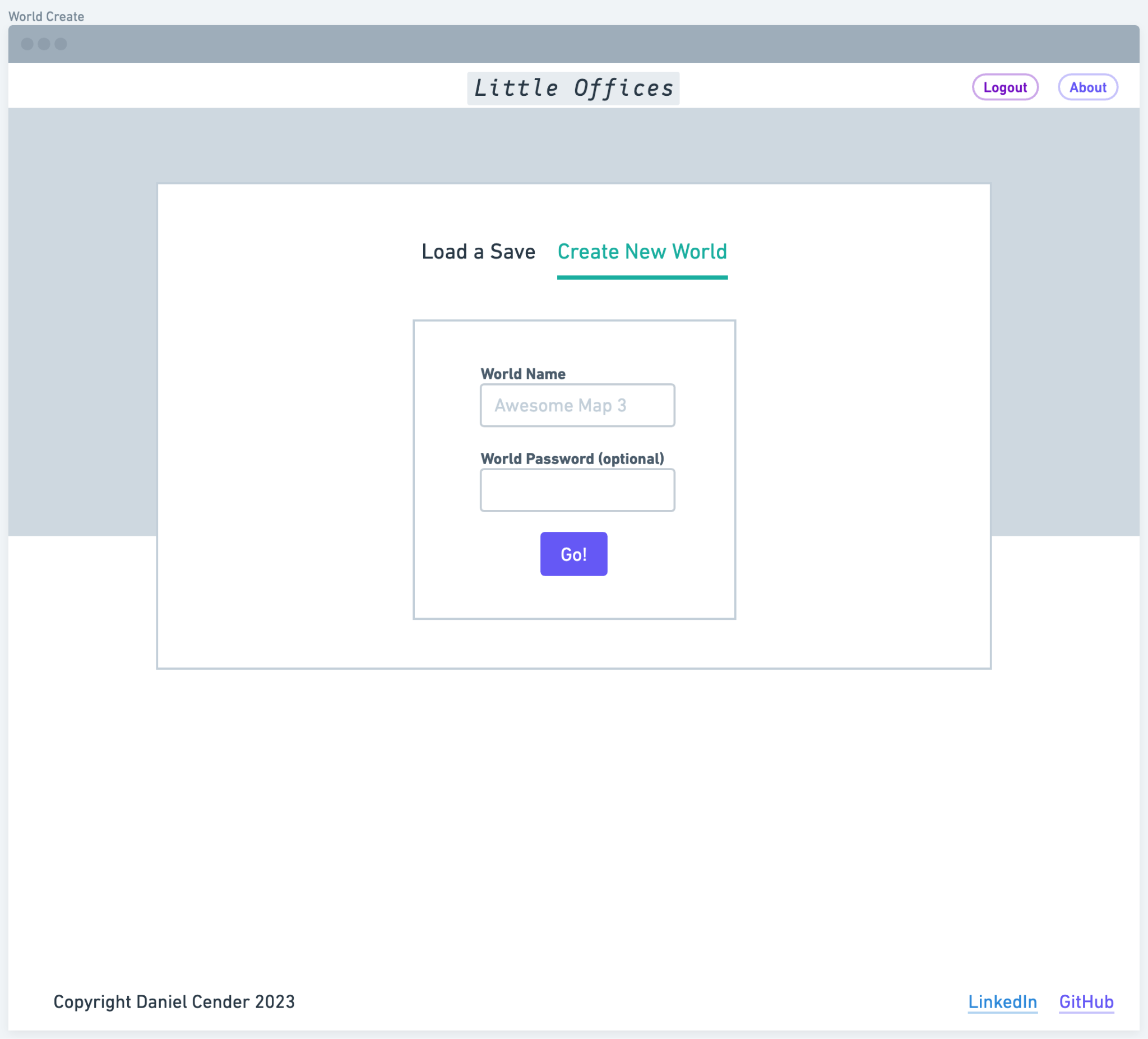
### *Login Page:*



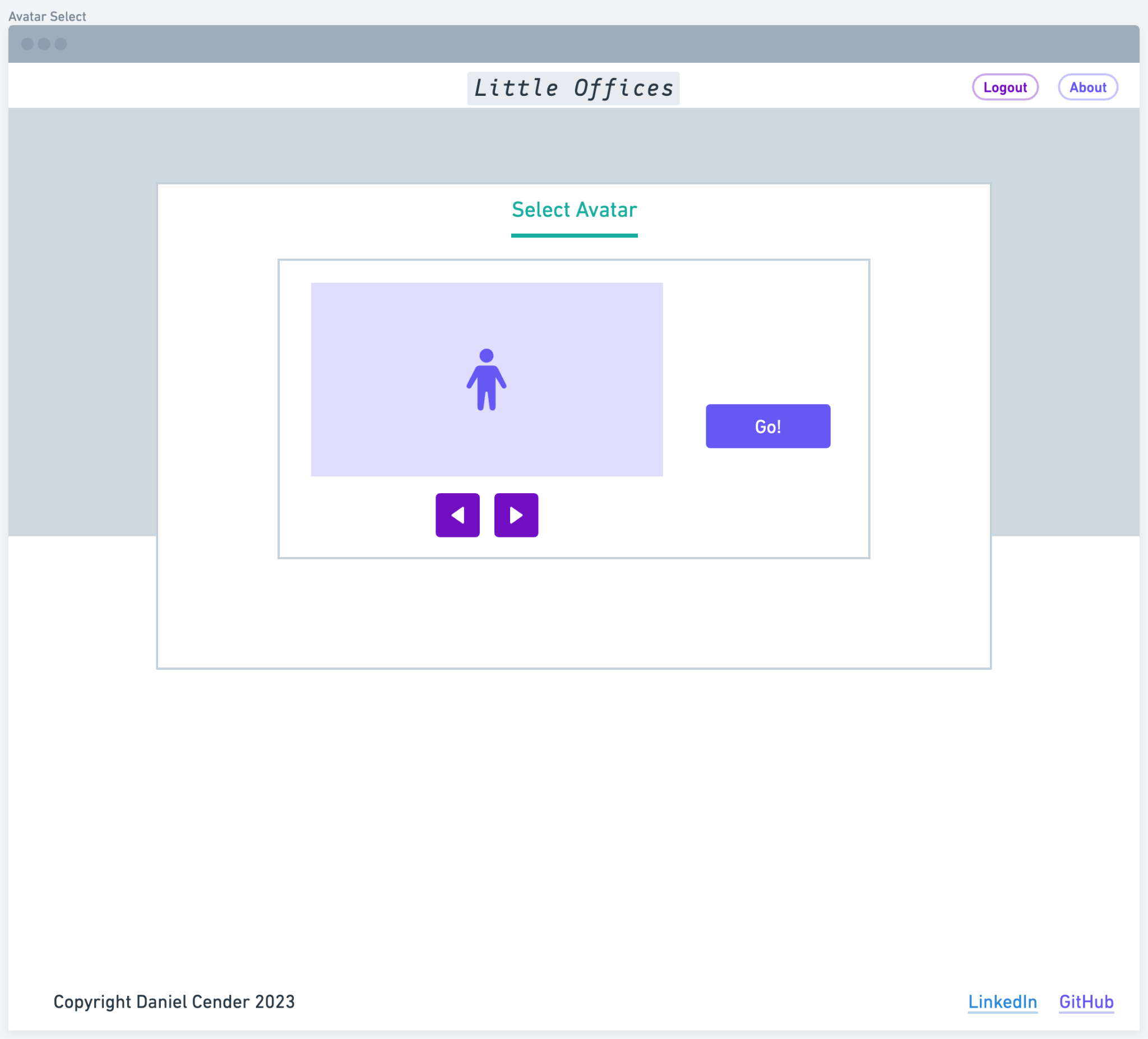
### *World Select:*



### *Create World:*



### *Avatar Select:*



### 

### 

### 

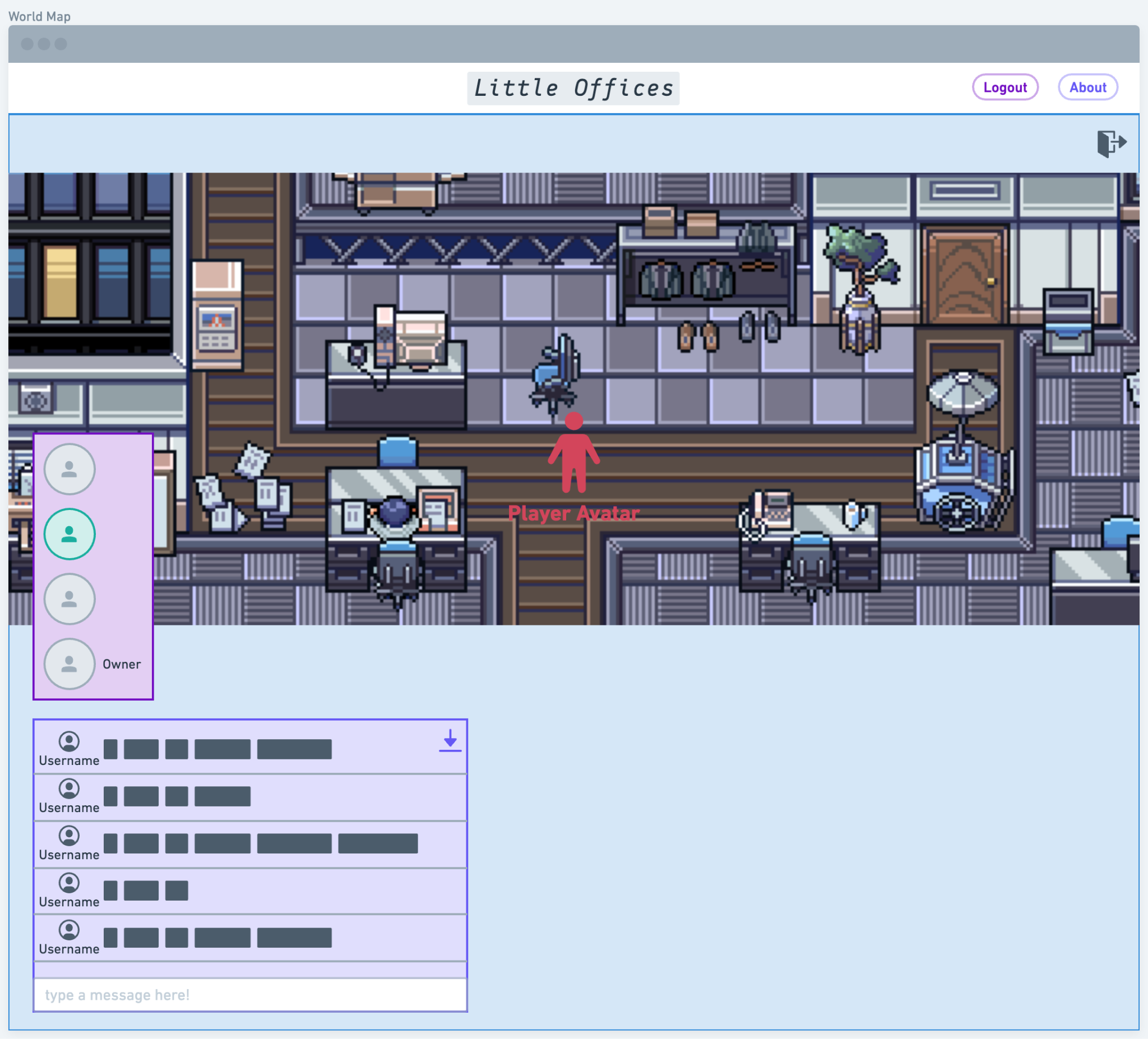
### 

### 

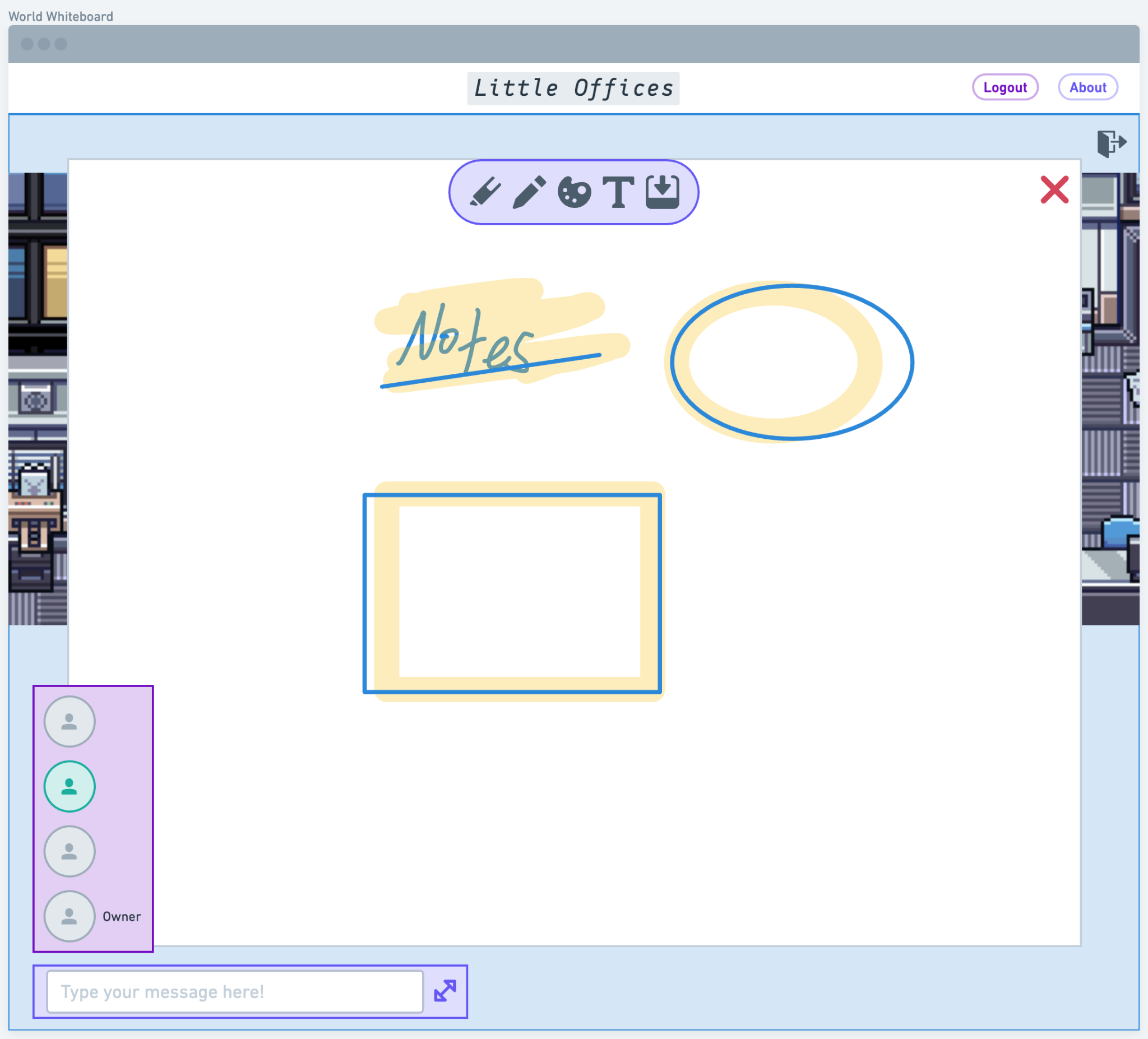
### 

### *World Map:*

The World Map is an approximation of where the different UI elements will appear in the main app view. Ideally, the user avatars on the left will be highlighted when that user is speaking over the voice chat.



### *World Whiteboard:*



# **SECURITY**

## User Roles

The application has just two distinct roles which users will embody. Each World can only have one World Owner. They have full administrative editing capabilities for the World object, including for full deletion. A World Attendee is anyone who’s joined the World, besides the World Owner. Logically, a World Owner has the same permission schema as a World attendee, but with a few extra permissions.

In a more full-fledged application, a world owner might be able to appoint moderators, or administrative users with elevated permissions. In this scenario, if a World Owner wants to “kick” any users from the current World, they’d either have to set a World password or just create a new World space with the remaining World Attendees.

## 

An initial diagram illustrates the roles’ layered design a bit more visually:

## 

# REPORTS DESIGN

This application will not produce any externally-accessible reports. The only data generated by users which might be applicable for reporting would be a log of the World messaging feature. There are no plans to allow message exports, however.