

CO



COLORADO SCHOOL OF MINES
EARTH • ENERGY • ENVIRONMENT

CSCI 370 Final Report

The Arthur Lakes Cartographers

Cammie Olsgard
William Vangbo
Isaak Velasquez
Weston Welshons

Revised November 12, 2023

CSCI 370 Fall 2023

Prof Kelly

Table 1: Revision history

Revision	Date	Comments
New	August 31 st , 2023	<p>Modified Sections:</p> <ul style="list-style-type: none"> I. Introduction II. Functional Requirements III. Non-Functional Requirements IV. Risks V. Definition of Done VI. Team Profile <p>References</p> <p>Appendix A—Key Terms</p>
Rev – 2	September 17 th , 2023	<p>Modified Sections:</p> <ul style="list-style-type: none"> I. System Architecture
Rev – 3	October 22 nd , 2023	<p>Modified Sections:</p> <ul style="list-style-type: none"> I. Non-Functional Requirements II. System Architecture III. Software Test and Quality IV. Project Ethical Considerations <p>References</p> <p>Appendix A—Key Terms</p>
Rev - 4	November 12 th , 2023	<p>Modified Sections:</p> <ul style="list-style-type: none"> I. Functional Requirements II. System Architecture III. Software Test and Quality IV. Future Work V. Lessons Learned
Rev - 5	December 10 th , 2023	<p>Modified Sections:</p> <p>All Sections</p>

Table of Contents

I. Introduction	3
II. Functional Requirements.....	6
III. Non-Functional Requirements.....	6
IV. Risks	7
V. Definition of Done.....	7
VI. System Architecture	7
Technical Issues.....	7
System Design	8
VII. Software Test and Quality	11
Testing Plan:	11
Testing Results:	13
Test results for API responses:.....	13
Performance Testing:	14
Library Staff Test:	14
VIII. Project Ethical Considerations	15
IX. Project Completion Status	15
X. Future Work	16
XI. Lessons Learned	17
XII. Acknowledgments.....	17
XIII. Team Profile	18
XIV. References	19
Appendix A – Key Terms	20
Appendix B – Website Maintenance.....	21

I. Introduction

The purpose of this project was to develop a new interactive map for the Arthur Lakes Library that could visually display study room availability and how occupied open study spaces were. This new map aimed to resolve issues students and faculty had accessing important information and services related to the library.

The Arthur Lakes Library is one of Mines' many resources available to students and faculty. The library has over 10 open study spaces and a dozen bookable study rooms spread out over 3 floors with a mezzanine on the first and second floors. Unfortunately, resources for checking room bookings and study space availability were spread out over two services. Both services were hard to parse for students not already familiar with the layout of the library. This, combined with the lack of high-quality maps of the library available online, made accessing library resources challenging for new and old students alike.

Previously, the library utilized two different services for displaying study room availability and overall study space crowd levels. The former was done through a LibCal application, which can be found at the following address: <https://library.mines.edu/services/rooms/>. This application interfaced with the LibCal API, which managed study room reservations at the Arthur Lakes Library. The interface for the application currently available is shown in Figure 1:



Figure 1: Arthur Lakes Library Study Room Booking Interface

How crowded different areas of the library were could be determined using the Waitz app, which was available as a mobile app or online at the following address: <https://waitz.io/mines>. This application interfaced with the Occuspace API, which used data from sensors placed throughout the library to estimate how crowded a given area is. The user interface for the Waitz app is pictured below in Figure 2:

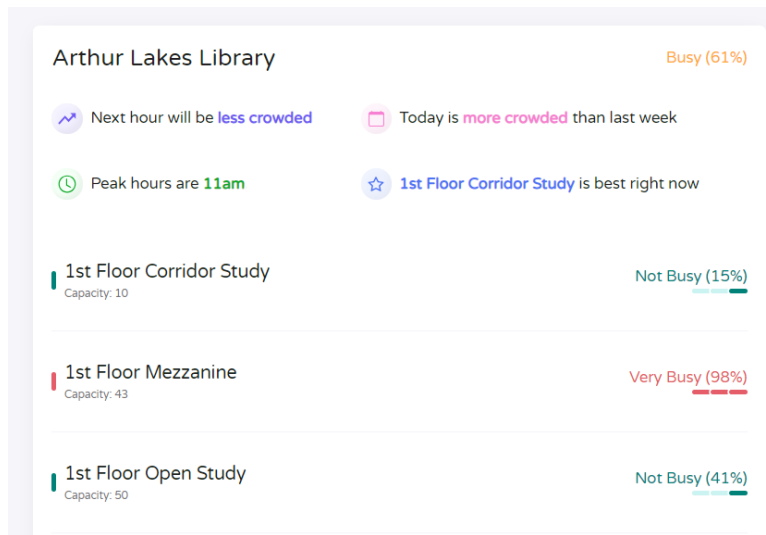


Figure 2: Open Space Availability Through the Waitz App

As can be seen in Figure 1 and 2, both applications were text based and did not clearly indicate where in the library they were referring to. Students seeking to book a study room may have struggled to find their room given the lack of high-quality floor plans currently available. These floor plans also did not explicitly show where study rooms or open study spaces were. The old floor plan for floor 2 is pictured in Figure 3:

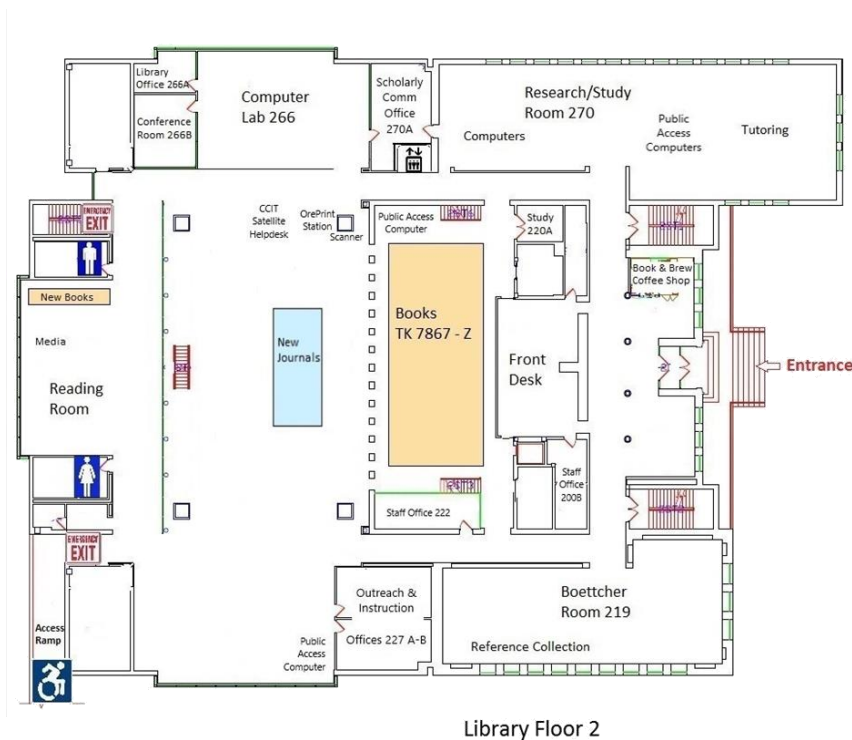


Figure 3: Arthur Lakes Library Floor 2 Plan

To use this map as an example, floor 2 of the Arthur Lakes Library had an open study space referred to in the Waitz app as “Floor 2 Open Study.” Newcomers at the library had no way to visualize where this was in the library, and may have even mistakenly identified the area labelled “Research/Study Room” as the “Floor 2 Open Study” area in the Waitz app, despite this room being referred to as the “Tutoring Research Room” in

the app. The inconveniences compound and discourage students new to the Arthur Lakes Library from taking full advantage of its resources.

Stakeholders for the project included Mines students and faculty who utilized the library's services or who may have wanted to use these services in the future. We also considered the librarians who wanted to see the continued success of the library as our stakeholders. Looking forward, we wanted to ensure that the Arthur Lakes Library was delivered a website that they could continue to maintain and expand upon in the future.

Adam Olsen is a Digital User Experience Librarian and served as our team's primary contact at the Arthur Lakes Library. Over the course of the semester, other staff members at the Arthur Lakes Library were able to give our team additional feedback on the direction of the project.

II. Functional Requirements

Our final product was delivered as a webpage. The following requirements guided us throughout the project:

- The website had to display a map of the Arthur Lakes Library.
 - o Because there are multiple maps representing each floor of the building, the user must have the ability to switch between the different maps without scrolling.
- The map had to show the current occupancy of each area in the library directly on the map.
 - o These occupancy metrics had to reflect the same metrics currently available on the Waitz app currently available at the following address: <https://waitz.io/mines>.
- The map had to show the location of study spaces and whether they are occupied on the map.
- The website had to allow library users to book study rooms directly from the map.

The final bullet point for our requirements was ultimately found to be out of scope for our project, and was thus replaced with the following requirements:

- The website had to display future room booking information for study rooms throughout the library.
- Highlighting a study room on the map would highlight that same room in the future bookings table, and vice versa.

In addition to these functionalities, the map also had to be accessible in accordance with the Web Content Accessibility Guidelines (WCAG) 2.2. The WCAG 2.2 discusses several guidelines about building websites that meet their 4 principles of website being perceivable, operable, understandable, and robust [1]. To ensure we met some of these guidelines on accessibility, we tested our website using the WAVE suite of tools produced by WebAIM. These tools allow developers to spot accessibility concerns that they may have been aware of otherwise by examining the HTML code used by a website [2]. Additionally, since our map would convey a lot of visual information, we had to use colorblind friendly palettes and textures to ensure people with protanopia, deuteranopia, tritanopia, and other visual impairments could access the site [3].

III. Non-Functional Requirements

The following non-functional requirements guided us throughout the project:

- The interactive library map had to be hosted and accessible via Arthur Lakes' Library's website, which utilizes Wordpress as its content management system.
- The page itself had to be written in JavaScript.
- The map had to utilize information from the Occuspace and LibCal APIs to determine the occupancy status of study rooms and areas.
 - o These API calls had to be done server-side to preserve the privacy of LibCal users and the security of both API keys. Due to this restriction, and since we had to keep our infrastructure within the Arthur Lakes Library Wordpress CMS, we had to use the PHP snippets Wordpress plugin for our API calls.
- The map had to be functional across different browsers and devices. Devices include personal and school computers, phones, and tablets.

IV. Risks

If incorrectly configured, users interacting with the interactive library map could receive false information about space availability within the Arthur Lakes Library. This could have made reserving spaces and finding a spot to study difficult for some students, rendering the project unhelpful. We mitigated this risk by frequently testing the map and ensuring that all the correct data was displayed on the map.

The interactive library map could have broken some functionalities of the Arthur Lakes Library website if improperly uploaded. The likelihood of this risk was low since the interactive map did not need to replace previously built features and exists on its own page. Any negative effects on user experience would affect students and faculty who utilize the library's website. This risk was mitigated by the team doing our coding and work on the Arthur Lakes Library developer website, which is not open to public use.

Improper API configuration could have resulted in users being able to take advantage of the LibCal room booking system. This risk was only present with the LibCal API, as one could send 'POST' requests to the LibCal API if they had the key, unlike the Occuspace API. Attempting to resolve these potential issues with the LibCal API could have made it difficult or impossible for students to make room reservations for a short period of time. We mitigated this risk by moving our API calls to Wordpress server-side PHP calls, preventing normal users from discovering the API key.

V. Definition of Done

Our final deliverable was a webpage integrated into the school library website system that displays an interactive map showing density of occupation and which rooms are booked for studying in real time. Beneath our map, we had to display information about future room booking information at the library.

The library map had to be accessible in accordance with Web Content Accessibility Guideline (WCAG) 2.2 recommendations. Information on the map conveyed using color needed to utilize accessible color schemes and the final site was tested for accessibility using the WAVE suite of tools from WebAIM. Additionally, our map had to scale properly on mobile devices and remain accessible to mobile users.

The client ensured that the map displayed the correct density information in the correct location and that the correct rooms were marked as reserved or open for booking. Additionally, all our interactive elements, including changing floors and viewing information about a specific area or room, had to work correctly and intuitively.

The product was delivered via the Arthur Lakes Library Wordpress page. It was delivered with a GitHub repo to hold SVGs of the map as well as a document containing information on how the code fits together and where different media is stored. This document contained information on how to expand the map in the future. The text of this document can be found in Appendix B. A downloadable version of this document, along with an exported version of the site itself and all our PHP snippets used, was sent to our client.

VI. System Architecture

Technical Issues

The system architecture of the project needed to overcome several obstacles throughout the development of the website. The first issue we ran into involved getting information from both of our APIs. The Occuspace API

was relatively easy to access, but the LibCal API had more security measures in place which made it harder to access and harder to test after we had gained access.

Next, around the midpoint of the semester, we had to pivot our plans for the system architecture to ensure that API calls were all being done server-side for reasons that will be elaborated on later. This pivot required us to gain more privileged access to the Arthur Lakes Library’s developer site. The overall structure of the system became more complicated after this pivot, requiring us to write additional documentation on where all the parts of our project existed within Wordpress.

System Design

The design of the Arthur Lakes Library map website is best described by an architecture diagram alongside wireframes of the website.

Figure 4 depicts all the different components that communicate with the interactive map website.

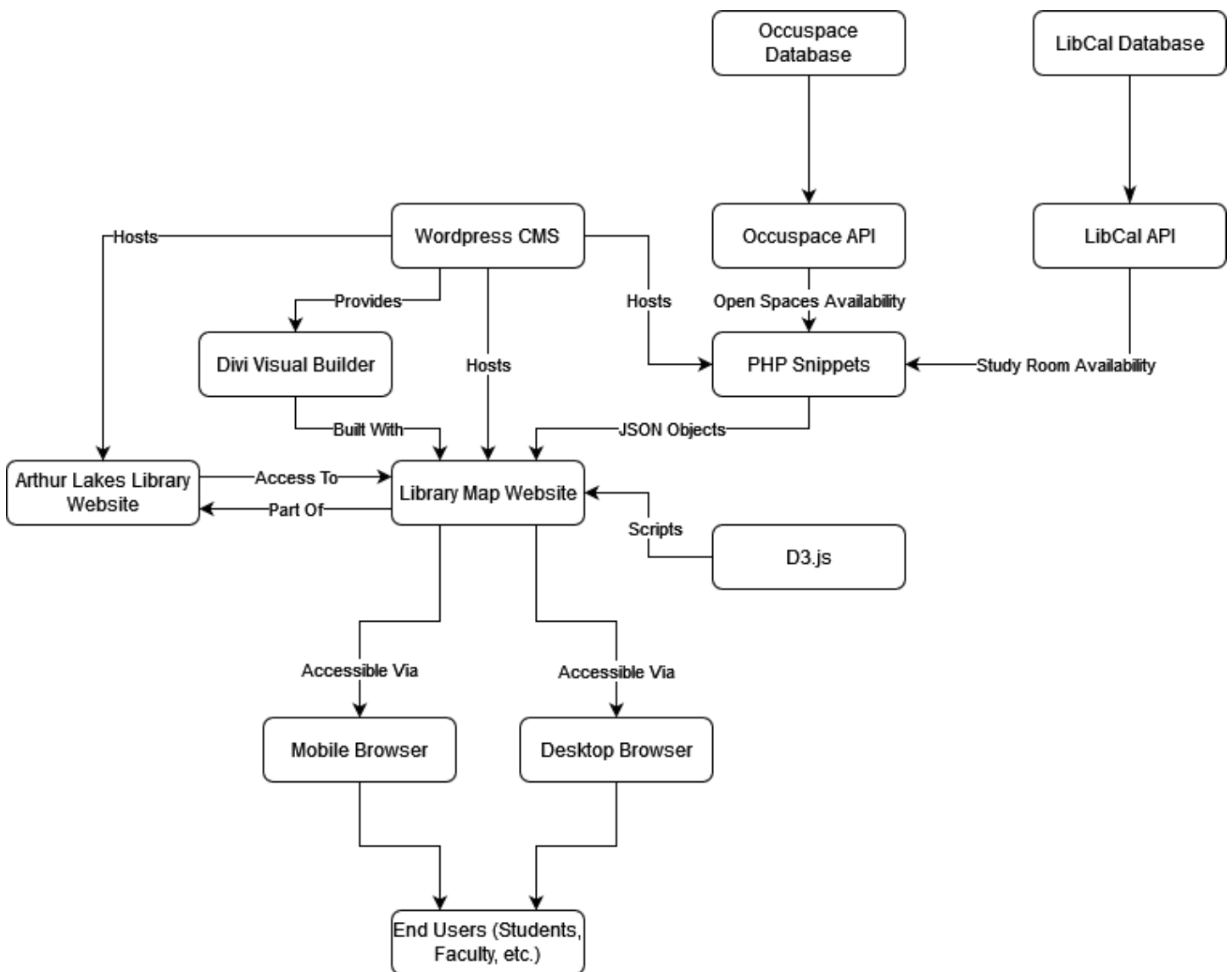


Figure 4: Library Map Architecture Diagram

First, there were the Occuspace and LibCal databases, which stored information about open space availability and study room available respectively. The library map website had to be able to receive information through the LibCal and Occuspace APIs. Note that the low-level specifics about how Occuspace and LibCal store and process data were out of the scope of our project and not relevant to the overall architecture.

Next, the library map website used an external Javascript library, D3.js, to aid in data visualization. This library was chosen as D3.js specializes in data visualization and made it easier to display information about room availability and how occupied open study spaces are.

The website was built using Divi, a Wordpress theme that allows one to use a visual editor to build a website within Wordpress. Divi was already in use by the Arthur Lakes Library for the rest of their website infrastructure, but in our project Divi was especially helpful for creating a responsive front-end user experience, meaning that the components of the map would scale properly to fit different devices. This was important to our project as our end-users would access the website using a variety of browsers on both mobile and desktop.

The library map website had to exist within the Arthur Lakes Library's current website infrastructure. This would allow for easy access between the library map and the rest of the library's website services. On a technical level, this meant that the library map would be hosted inside of the library's Wordpress site and that the content of the library map had to function appropriately when hosted on Wordpress. To preserve the privacy of users booking rooms through LibCal and the secrecy of API keys, all the API calls were done using PHP snippets uploaded the Arthur Lakes Library website. The PHP snippets work by using AJAX requests and converting data received from the Occuspace and LibCal APIs into a JSON object, which could then be parsed and visualized by the library map website. API calls were done on the server side so sensitive information would not be revealed to users of the website.






The following 2 figures contain wireframes that we used throughout the project to help guide development on the map website.

FLOOR SELECTION

Floor 2 Selected!

- Floor 3
- Floor 2 Upper
- Floor 2
- Floor 1 Upper
- Floor 1

MAP LEGEND

-  Available Study Room
-  Booked Study Room
-  Not Busy
-  Busy
-  Very Busy



Click [here](#) to reserve a study room or a conference room.






Time	6:00 PM	7:00 PM	8:00 PM	9:00 PM	10:00 PM
220A					

Figure 5: Desktop Wireframe



Figure 6: Mobile Wireframe

The components of the map stayed the same across different devices. There was a radio button for floor selection, a legend, a visual display of the currently selected floor, and a table at the bottom of the page containing future booking information. Above the current floor display there was a line of text describing the availability of interactive areas on the map, which could be hovered over. These components would align vertically when accessed on a mobile browser. Note that the colors utilized by the map legend were picked to be colorblind accessible, but that different textures were also used to help convey availability information to users.

VII. Software Test and Quality

Testing Plan:

API testing focused only on how our code interacted with the APIs and handled server and client-side issues, for it was expected that the respective companies that provide the APIs have properly maintained and tested the APIs that our code used. The website should maintain basic functionality and still display the map regardless of the functionality and status of the client's APIs. The tests mainly consisted of identifying unexpected behaviors of the webpage when data from the APIs are missing, and making sure that the core functionality of the webpage is not dependent on the success of API calls coming from an outside source. These API tests ensured that many of our non-functional requirements were met.

Tests that we ran include:

- Test API authentication errors by testing the code with expired and invalid API keys to ensure that the site remains functional when these errors occur.

- Make test cases for 400 and 500 level API responses to ensure that client side and server-side issues with the APIs won't affect the overall functionality of the webpage.
- Tests that ensure that the scripts call and process the real API calls correctly by creating dummy JSON objects that contain known and expected data, and then making assertions with the data received from the real calls.

The following figures show a sample JSON object returned from the Occuspace API, available to view on their website [4], and a JSON object returned from the LibCal API for the Arthur Lakes Library:

```
{
  "message": "OK",
  "data":
  {
    "id": 2,
    "name": "Office Building",
    "count": 100,
    "percentage": 0.20,
    "timestamp": "2020-02-15T15:30:00.000Z",
    "isActive": true,
    "childCounts":
    [
      {
        "id": 3,
        "name": "Office 1st Floor",
        "count": 70,
        "percentage": 0.18,
        "isActive": true
      }
    ]
  }
}
```

Figure 7: Occuspace JSON Response.

```
{
  "id": 36126,
  "name": "120B-M",
  "booking_details": [
    {
      "bookId": "cs_j9W86aCP",
      "status": "Confirmed",
      "fromDate": "2023-10-20T09:00:00-06:00",
      "toDate": "2023-10-20T12:00:00-06:00"
    },
    {
      "bookId": "cs_P2PKr9IK",
      "status": "Confirmed",
      "fromDate": "2023-10-20T12:00:00-06:00",
      "toDate": "2023-10-20T13:00:00-06:00"
    }
  ]
}
```

Figure 8: LibCal JSON Response.

“Message” contains a simple response for whether everything worked correctly or not. “Data” holds a nested JSON for each of whatever it is we requested. The example above represents live data and thus contains a percentage of activity from the moment the API was called. The LibCal responses follow a similar JSON format, but contain study rooms with details regarding the booking status for the current day.

Our team engaged in consistent code reviews (at least once a week) where we looked over the code written for the project and ensured that it was the best quality we can manage. Since we were working with several different technologies which some of us were more familiar with than others, these code reviews provided us with the opportunity to determine how reasonable and efficient our solutions created for the project were. If one of us had trouble understanding what a segment of code contributed to the larger project, code reviews

allowed us to determine whether that segment of code was necessary or could be improved upon. If it was necessary and the best solution to the problem, but was still hard to understand, we marked these segments for further documentation. This is relevant to the quality of our product as we wanted the Arthur Lakes Library staff to be able to understand how the interactive library map works and why we made certain decisions, so they can further expand on the map in the future.

Accessibility testing for the map was done using WebAIM's WAVE suite of tools. These tools are designed to allow one to easily spot some WCAG violations [2]. The following is a list of a few examples of the kinds of accessibility issues that have been spotted by WAVE during the testing of our map:

- Low contrast colors
- Missing form labels
- Missing alternative text for images

Testing was done regularly using WAVE to ensure that we had not added a new component to the map that resulted in accessibility issues that would be difficult to spot further down the line. Tests that can't be done using WAVE were done manually using the relevant tools to bring the library map website up to WCAG standards. For example, testing for color blindness was done using Dr. David Nichols' tools for testing for protanopia, deuteranopia, and tritanopia colorblindness [3].

For the week of 11/13 – 11/17, we scheduled an opportunity to have library staff look over the new website and provide feedback. The goal of this portion of the testing plan was to gauge how helpful the interactive map of the library would be to the library staff. We wanted to ensure that the user interface of the library map was easy to understand, included relevant information, and was generally pleasant to look at and use. Since these experiences are subjective and may vary from person to person, the test was conducted by multiple staff members. Getting the opinions of many people was important for final quality considerations and delivering the best possible product. These tests ensured that the project was on track to deliver on all the functional requirements for what the website is supposed to do and ensured that we were meeting the non-functional requirement of having the page be accessible across different devices.

Testing Results:

Test results for API responses:

The validity of successful API responses was determined by our knowledge of the number of study rooms and areas that are in the Arthur Lakes Library. The number of elements in the JSON responses from the Occuspace and LibCal APIs were compared to the actual number of areas and study rooms. While the results of our tests were successful, and the number of elements in the responses were accurate, our front-end code still needed to account for responses that fail these tests in the event of missing data from the API.

400 and 500 level API responses were tested by passing dummy API keys and documenting front end behaviors when errors occur in the server-side PHP scripts. It was found that these responses only affected the components that relied on the respective API that was being tested. For example, if the Occuspace API call got a bad response, the features that rely on the LibCal API remained fully functional and vice versa. The components relied on the APIs that were tested lost all functionality and the SVG elements that displayed them became static.

Missing data from the API responses were tested by manually removing individual elements from the JSON objects that were returned from the LibCal and Occuspace API calls. Missing data resulted in the same behavior that was seen when testing for failed responses. To mitigate this, changes needed to be made to the front-end code to account for these potential API issues and maintain a user-friendly interface and

intractability. For example, if one study room was missing in the JSON response from the server, the map should still display all other study rooms correctly, and the respective SVG element for the missing room should still have an interactive element and inform the user that the data for that room is missing. Changes to the front-end code involved adding separate 'mouseenter' and 'mouseleave' events to SVG elements whose IDs did not have a corresponding entry in the JSON data. Aside from this issue, if the server sent back an invalid JSON as a response, the site would lose all functionality and the map would not display. This was resolved by rewriting the code to check if the response was a valid iterable object before attempting to parse it.

Performance Testing:

As the project became more complicated and the SVG file grew in size, something changed that caused a noticeable slowdown of the site. Specifically, there was a significant delay between hovering over an area and the map responding by darkening the area and changing the text. To determine the cause of this, we used the performance testing feature in Google Chrome's developer tools, which records the user interacting with the site and keeps track of how long certain actions take. It was found that much of the lag came from the 'Hit Test' category, which are triggered by 'mouseover' and 'mouseout' events. A few changes were implemented to increase the speed of the site, such as adding the 'content-visibility: hidden' attribute to the floors of the map that are not currently visible in order to reduce the number of events that needed to be handled. The 'mouseover' event listener was also changed to a 'mouseenter' listener, because the latter is only triggered once when it enters the object, whereas the former can be triggered multiple times as it enters sub-objects. Although these changes may have helped slightly, it was found that some lines were being drawn not as solid lines, but as dashed lines with a very large number of dashes, and this was the actual source of the lag.

Library Staff Test:

At the end of the 11/13-11/17 week we received an email from our client containing what he believed to be the most important feedback from the staff test. The exact feedback we received was:

1. Change the "upper" on the floors to "M."
2. The radio buttons are a little hard to select on mobile. Can we put more space between them, and possibly make them larger?
3. When changing floors, let's have the top "busyness level" text default to an area on that floor (say, the largest area on that floor).
4. Add scanner icons on floors 1 and 2.
5. Add on hover highlighting for the study rooms so that the study room available on the table highlights when hovering over the study room on the map (and vice versa).

We were able to successfully address each of these points for our final deliverable. The first point was a simple name change on the button labels. For the second point, we modified our code so that both the text and the button itself were clickable, thus giving users a larger, more forgiving area to interact with. To address the third point, we added a new section in the map code that replaced our original "Hover an area to see more details" text with instead the text of the largest area on the respective floor, without waiting for users to first mouse over it. The fourth point was addressed by editing the SVGs and uploading the new ones to the Github. The final point was completed by adding more code and changing how we were using some of the objects within our map.

VIII. Project Ethical Considerations

Since the library map displays information about crowds and room bookings, considerations had to be made to ensure the privacy of people who use the library's facilities. This was relevant to principles 1.02 and 3.12 under ACM guidelines [5]. Different considerations had to be made regarding the two APIs that will be accessed. Looking first at the Occuspace API, Occuspace sensors do not utilize camera data and no personal information is ever stored within the system. When examining the existing integration of LibCal's services on the Mines' website, while room bookings are shown throughout the day, no identifiable information is available about who booked which room. This data is stored by the library and made available in person as printed booking times are available to view outside of several study rooms in person. For our library map, our goal was to ensure the same level of privacy for students booking rooms in the library that was offered by the existing LibCal app available on the Arthur Lakes Library website. If the LibCal API secret were to be stored insecurely, personally identifiable information relating to room bookings could be revealed, which is why all API calls had to take place on the backend of the site.

Next, we needed to ensure that our project was properly documented in accordance with principle 3.08 under ACM guidelines [5]. We found this to be especially relevant for our project as up-to-date information on the library maps is crucial in ensuring the future efficacy of our website. For example, if the library were to permanently close some study rooms or track more open study spaces, we wanted to ensure that our map can be easily updated by future librarians who may only have some JavaScript knowledge.

In accordance with principle 1.07 under ACM guidelines, the library map must be accessible [5]. Relevant disabilities for our project included color blindness and other difficulties related to reading text and images. Accessibility features were tested using the WAVE suite of tools from WebAIM, which allows one to easily track issues like missing alternative text for images. Colors used by our website were tested for readability for color blindness using tools provided by Dr. David Nichols [3].


Finally, in accordance with ACM principle 3.05, we ensured that the library map website used the appropriate tools to complete all its tasks [5]. This meant that our group had to write clean code and put care into constructing elements of our system design to ensure that we always followed best practice. This was helpful to ensure the continued functionality of the open spaces library map and that librarians in the future could easily be able to improve or augment the functionality finalized by the end of the semester.

IX. Project Completion Status


At time of writing, the open spaces map is fully functional and available on the Arthur Lakes Library site at the following address: <https://library.mines.edu/open-spaces-map/>. The map is also being promoted on the landing page of the Arthur Lakes Library site, seen below in Figure 9.

SEARCH


[Databases A-Z](#) |
 [E-Journals](#) |
 [Advanced Search](#) |
 [Can't Find It?](#)




**RESEARCH
GUIDES**




**STUDY
ROOMS**



**MINES
REPOSITORY**



**SPECIAL
COLLECTIONS**



**GADGETS TO
GO**


Floor 2 Selected

- Floor 3
- Floor 2 M
- Floor 2
- Floor 1 M
- Floor 1

MAP LEGEND


- Available Study Room
- Booked Study Room
- Not Busy
- Busy
- Very Busy

BOETTCHER ROOM - BUSY (57%)



CAN'T FIND IT?

Need an article or book chapter from another institution? Arthur Lakes Library can borrow from over 10,000 libraries worldwide to get you the material you need for free.




NEW LIBRARY MAP

Our new library map shows which study areas and bookable rooms are available in real time!

NHV RESEARCH HELP

Tips, resources and research help for students taking HASS 100.



CAREER TOOLKIT FOR GRADUATE STUDENTS

Workshop series for tomorrow's researchers




Figure 9: Screenshot of the Arthur Lakes Library Website

We were successfully able to integrate the feedback we received on the project and resolve most of the other bugs and issues we ran into during the development of the project. We have documented the process for expanding the map in the future, this documentation is included in Appendix B.

Moving the map to the library site students are currently able to access will be relatively simple due to import/export functionality integrated into the Divi builder and the PHP snippets plugin used. We have documented exactly where the website loads in external resources from, such as the SVG of the map itself, which should make expanding on these resources easier for future developers.

X. Future Work

The library has some bookable rooms that are not necessarily study rooms, including a conference room on the 2nd floor. We chose not to integrate these rooms into the map since study rooms were always our top priority, but future developers should find adding new study rooms or conference rooms to the map to be relatively easy.

Future work on the Arthur Lakes Library open spaces map should involve finding ways to better integrate booking a study room directly into the library map. This will require integration with Mines Single-Sign-On, integration with LibCal's terms and conditions, and building a user interface for viewing your personal upcoming bookings. Recreating these services would require a large amount of development time, so future developers should work to assess how impactful this service would be.

Library staff have considered placing tablets throughout the library that the library map website could run on. Future work may involve optimizing the map experience for these devices specifically. Ultimately, without access to these tablets, we were unable to optimize the website for these devices.

XI. Lessons Learned

Our first major lesson came during the process of investigating how our project would affect student privacy. There was a much higher potential for data leaks than we originally anticipated. Working with sensitive data was not something we had all experienced before, and this project gave us a good foundation for how to think about data and ways that a system can be built to better maintain privacy. In future projects, we expect to start the design process with privacy in mind, as trying to add these features later into the process resulted in us losing some amount of development time.

During the development process, we were also able to experience designing a webpage mostly from scratch, and how the front-end and back-end tools of web development interact. We gained a lot of experience with web development within Wordpress specifically. Our front-end experience within Wordpress focused on using the Divi builder and other tools provided by Wordpress to make a visually appealing website. Looking at our back-end experience, we learned a lot about the limits of what a Wordpress page can do on its own and how developers respond when given situations where the tools available are unable to complete a given task. For us, our main struggle with Wordpress revolved around our ability to integrate PHP code that could perform server-side API calls. Ultimately, we used a plugin created by other developers that allowed us to host PHP snippets on the Wordpress site. Going forward, instead of struggling to use a tool for purposes it was not meant for, we have learned to explore different tools available that may better suit our development needs.

XII. Acknowledgments

Our team would like to thank Adam Olsen, our primary contact at the Arthur Lakes Library, for directing us through this project by providing helpful feedback and being communicative and understanding. We would also like to thank the Arthur Lakes Library staff for their feedback throughout the project. This feedback was essential to ensure that our final product did not have any major issues before release. Finally, we would like to thank our advisor, Dr. Kelly, for her feedback on the project and her general advice throughout the semester.

XIII. Team Profile

Cammie Olsgard

Senior - Computer Science

Hometown: Castle Rock, CO

Work Experience: Student Assistant at the Colorado School of Mines ORA, Senior Counselor at Silicon STEM Academy

Hobbies: Music, video games, cooking/baking

I am excited to gain more web development experience and make something that my fellow students can use for years to come.

William Vangbo

Junior - Computer Science

Hometown: Longmont, CO

Work Experience: Internship at Data Ductus, previous TA for CSCI101 Introduction to Computer Science, current TA for CSCI358 Discrete Math

Hobbies: Reading, painting, baking

I am excited to work with the Arthur Lakes Library team to make utilizing the library's services more accessible for Mines students.

Isaak Velasquez

Senior - Computer Science

Hometown: Denver, CO

Work Experience: Internship at Spectrum and gained soft skills from service industry experience

Hobbies: Guitar and music, rock climbing, camping, reading

I've always wanted to learn about front end web development, and I think this project is a great opportunity to do this.

Weston Welshons

Junior - Computer Science

Hometown: Houston, TX

Work Experience: None

Hobbies: Watching movies, music, video games

I am very excited to learn and get experience from this project, and to do something that impacts Mines.

XIV. References

- [1] "Web Content Accessibility Guidelines (WCAG) 2.2," [Online]. Available: <https://www.w3.org/TR/WCAG22/>. [Accessed 10 12 2023].
- [2] WebAIM, "WAVE Web Accessibility Tool," [Online]. Available: <https://wave.webaim.org/>. [Accessed 10 12 2023].
- [3] D. Nichols, "Coloring for Colorblindness," [Online]. Available: <https://davidmathlogic.com/colorblind>. [Accessed 10 12 2023].
- [4] "Occuspace API," [Online]. Available: <https://occuspace.io/api>. [Accessed 10 12 2023].
- [5] ACM, "ACM Code of Ethics and Professional Conduct," 22 6 2018. [Online]. Available: <https://www.acm.org/code-of-ethics>. [Accessed 10 12 2023].

Appendix A – Key Terms

The below table includes some key terms and acronyms referenced throughout the project.

Term	Definition
<i>AJAX</i>	<i>AJAX stands for Asynchronous JavaScript and XML. AJAX is a popular framework due to its ability to update data asynchronously, so a user won't necessarily have to reload a page to receive the most up-to-date information from a server.</i>
<i>WAVE</i>	<i>WAVE is a set of accessibility evaluation tools built by the WebAIM team at the Institute for Disability Research, Policy & Practice at Utah State University [2]. More information on these tools can be found here: https://wave.webaim.org/</i>
<i>WCAG 2.2</i>	<i>The Web Content Accessibility Guidelines (WCAG) 2.2 is a set of accessibility guidelines from the World Wide Web Consortium [1]. More information on the guidelines can be found here: https://www.w3.org/TR/WCAG22/</i>
<i>CMS</i>	<i>CMS stands for "Content Management System." Broadly, a CMS allows different users to update and compile the content of a website. Within the scope of this project, the Arthur Lakes Library utilizes Wordpress as its CMS.</i>

Appendix B – Website Maintenance

The following guide was developed to give the client detailed instructions on how to expand on the map in the future. This guide also provides a more detailed overview of how specific sections on the website map function.

Wordpress Site Overview

Divi Builder

Below is a screenshot of what the Arthur Lakes Library open spaces map looks like in the Divi Builder.



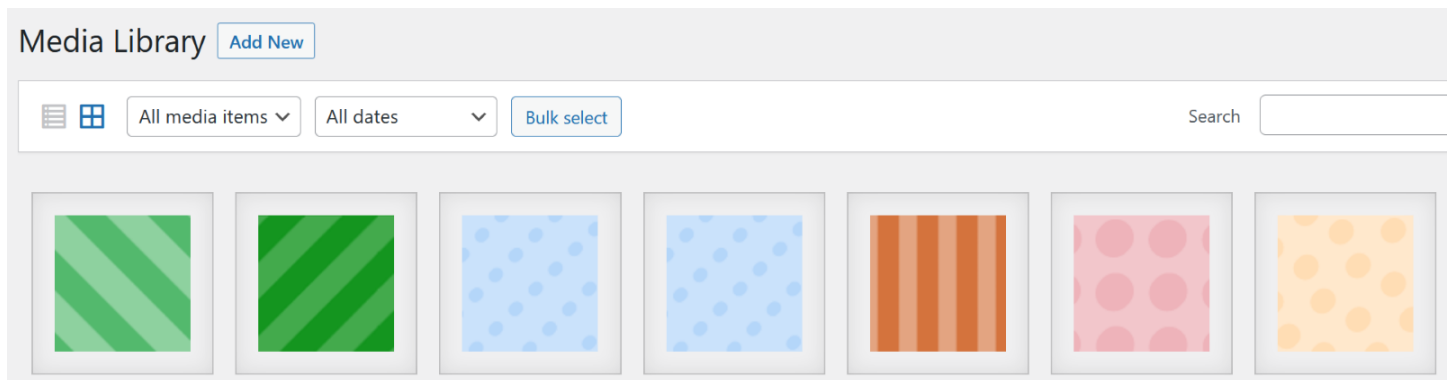
The left column in the first row contains the HTML used for the floor buttons, a title that sits above the legend of the map, and the HTML for the Legend itself. The Legend utilizes references to icons uploaded to the Arthur Lakes Library dev site’s media page.

The right column contains the scripts used by the site to get information from the LibCal and Occuspace APIs, change how the map displays according to the information received from these APIs, and change which floor is currently being viewed. The SVG of the map and the loading symbol are both loaded into the page from the Arthur Lakes Library’s github repository. This was necessary due to Wordpress being unable to host SVGs in the built-in media tab for security reasons. Note that the Floor Button Script also changes the text displayed above the floor buttons and writes the tables showing future availability.

The second row contains two static text elements along with an HTML element that contains a table referenced within the Floor Button Script to show future room availability.

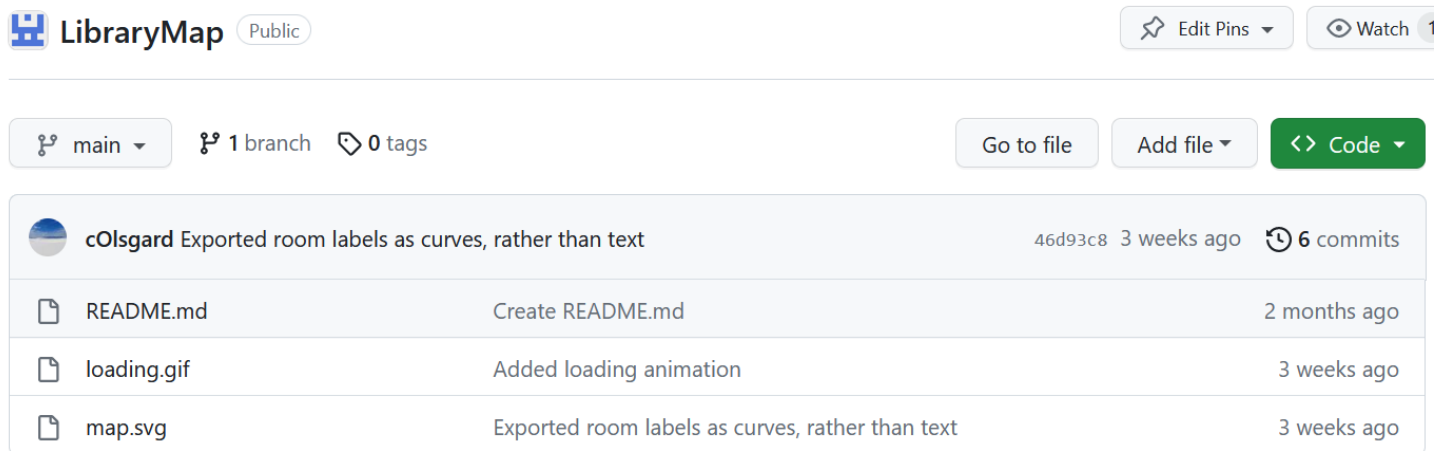
Media Tab

The media tab on the Wordpress dev site is used to host icons used in the map legend and in the future availability table. In the future, these icons may be moved to the GitHub repository or to the media tab of the Arthur Lakes Library website, but during our testing we found that our own locally hosted instances of the map website were able to reference objects in the media tab of the developer site with no issues.



Library Map GitHub Repository

The LibraryMap GitHub repository hosts a publicly available SVG of the Arthur Lakes Library map as well as a small gif to show that the library map is being loaded in since loading the data for the map can sometimes take a few seconds.



PHP Snippets

All API calls for the retrieval of LibCal and Occuspace data are done by PHP code snippets running on the server side and are hosted on the Wordpress site. The outputs of these snippets are accessed by the frontend

via JavaScript AJAX requests and are in the form of JSON arrays. It's imperative that the code for these snippets remain private and are hidden from the client at all times, for they contain API keys that will grant access to private student information if they are compromised. Documentation for the APIs can be accessed here:

- Occuspace: <https://occuspace.io/api>
- LibCal (admin privileges required):
https://mines.libapps.com/libapps/login.php?site_id=12831&target=%2Fadmin%2Fapi#s-lc-tab-endpoints_1.1

There are 4 PHP snippets located in the "Snippets" tab within the Wordpress developer site:

- **occuspace**
 - Defines a function that calls Occuspace APIs and constructs a JSON array whose objects correspond to each area in the Arthur Lakes Library.
 - This snippet utilizes GET API calls using PHP cURL requests to two different API endpoints. The first is made to '<https://api.occuspace.io/v1/locations>' to obtain all area IDs. These IDs are then used to obtain the real time data for each area by iteratively calling '[https://api.occuspace.io/v1/locations/\[INSERT ID\]/now](https://api.occuspace.io/v1/locations/[INSERT ID]/now)' with each area ID.
- **libcal_study**
 - Defines a function that calls LibCal APIs and constructs a JSON array whose objects correspond to each study room in the Arthur Lakes Library.
 - This snippet depends on the time_config snippet to obtain a time sensitive API token that's needed to retrieve the real time data for the study rooms.
 - This snippet utilizes GET API calls using PHP cURL requests to two different API endpoints. The first is made to '<https://libcal.mines.edu/1.1/space/items/5642>' to obtain all study room IDs. The IDs are then used to obtain the real time data for each study room by iteratively calling '[https://libcal.mines.edu/1.1/space/bookings?eid=\[INSERT ID\]](https://libcal.mines.edu/1.1/space/bookings?eid=[INSERT ID])' with each study room ID.
 - Most attributes obtained from the second API call are omitted from the final output, for they contain sensitive information that should not be returned to the client.
- **get_token**
 - Defines a function that calls the LibCal authentication endpoint to retrieve a new temporary time sensitive API token.
 - This snippet is called by the time_config snippet to ensure that new tokens are only generated when necessary.
 - This snippet utilizes POST API calls using PHP cURL requests to '[https://libcal.mines.edu/1.1/oauth/\[INSERT TOKEN\]](https://libcal.mines.edu/1.1/oauth/[INSERT TOKEN])' (using a permanent access token generated in the LibCal site) to obtain the temporary token.

- **time_config**

- Defines a function that returns the active temporary token and will periodically call the `get_token` snippet once per hour to generate a new temporary token once the previous one has expired.
- This snippet utilizes [Wordpress transient APIs](#) to store active temporary API tokens.
- It's imperative that new tokens are only generated once the previous one has expired (once every hour), for large volumes of token requests are against LibCal's policy. Note: this doesn't necessarily apply when testing so long as the volume of subsequent requests remains reasonably low.

PHP Snippet Usage

AJAX:

Wordpress requires functions defined within the PHP snippets to be explicitly registered as AJAX functions to be referenced in front end AJAX requests:

```
// This is so wordpress registers this function as an AJAX function
add_action( 'wp_ajax_libcal_study', 'libcal_study' );
add_action( 'wp_ajax_nopriv_libcal_study', 'libcal_study' );
```

Example usage in the `libcal_study` snippet.

libcal_study modification:

The data returned by the `libcal_study` snippet (JSON objects) was carefully chosen to protect student privacy. If more information needs to be added to the output, it's important to always consider that it will be visible to the client.

The code block below shows where the desired attributes to be returned by the snippet are specified by adding them to the `"$room_booking"` array. A detailed list of the attribute options can be found in the LibCal documentation.

```
50         foreach($dec_response as $booking){
51             $room_booking = array(
52                 "bookId" => $booking->bookId,
53                 "status" => $booking->status,
54                 "fromDate" => $booking->fromDate,
55                 "toDate" => $booking->toDate,
56             );
```

To add a new attribute, add the new attribute(s) to the `"$room_booking"` array on a new line following the format:

```
"new_attribute_name" => $booking->new_attribute_name,
```

Modifying the map

SVG attributes and hierarchy

The map of the library is stored on the Arthur Lakes Library GitHub page, specifically in the LibraryMap repository. This file is called map.svg, and any changes made to that file will be immediately seen on the live website. All elements in the map.svg file are contained within five groups, each representing a floor (or subfloor) of the Arthur Lakes Library. These groups have the "floor" class and an id of "floor-###", corresponding to the floor number plus 'M' if it is a mezzanine floor. Within each floor group there are subgroups, with their class names listed below:

- "background" – solid color background for the floor
- "spaces" – Occuspace areas
- "rooms" – Rooms that are not study rooms
- "study_rooms" – Bookable study rooms
- "stairs", "walls", "doors", and "shelves"
- "symbols" – Symbols for notable features, such as computers or scanners.
- "labels" – Labels for rooms that are not study rooms

If the map.svg file needs to be modified, a program must be used that can preserve its XML attributes on import and export. Adobe Illustrator and Inkscape should be capable of this.

Adding an Occuspace area

The scripts themselves do not need to be modified in order to add or remove an Occuspace area, however the map.svg file must be modified. Occuspace areas need to be added inside the "spaces" group for whatever floor the area is on. A 'space' is a group that has an ID of 'space-####', with the pound symbols being replaced by the ID of the space returned by the Occuspace API. This group must contain a path for the color fill, a clipPath of the same shape, and a group of lines that are clipped by the clipPath. The first two paths will likely need to be made from scratch, but the group of lines can be easily copied from other existing spaces, however the position of the lines might need to be changed.

Here is an example of what a new Occuspace area might look like. It should contain all of these elements, along with the exact classes and IDs shown below (the ID of the clipPath does not matter).

```
<g id="space-SPACEID">
  <path class="space_fill" .../>
  <clipPath id="_clip123">
    <path class="space_fill" .../>
  </clipPath>
  <g clip-path="url(#_clip123)">
    <g class="lines">
      <path .../>
      ...
      <path .../>
    </g>
  </g>
```

```
</g>  
</g>
```

Adding a study room

Study rooms must be added to the "study_rooms" group for that floor. A study room is a group that has an ID of 'study-#####', where the pound signs are replaced by the ID of the room in LibCal. This group must contain a path (or rect) for the color fill and a clipPath of the same shape, along with a group of lines that are diagonal ("open-lines") and a group of lines that are vertical ("closed-lines") being clipped by the clipPath. Along with these elements, there must also be separate paths and lines for when no data is available ("study_err").

Here is an example of what a new study room might look like. It should contain all of these elements, along with the exact classes and IDs shown below (the ID of the clipPaths do not matter).

```
<g id="study-ROOMID">  
  <rect class="study_fill" .../>  
  <clipPath id="_clip456">  
    <rect .../>  
  </clipPath>  
  <g clip-path="url(#_clip456)">  
    <g class="open-lines">  
      <path .../>  
      ...  
      <path .../>  
    </g>  
    <g class="closed-lines">  
      <path .../>  
      ...  
      <path .../>  
    </g>  
  </g>  
  <rect class="study_err" .../>  
  <clipPath id="_clip789">  
    <rect .../>  
  </clipPath>  
  <g clip-path="url(#_clip789)">  
    <g class="lines">  
      <path .../>  
      ...  
      <path .../>  
    </g>  
  </g>  
</g>
```

This is all that is necessary to add a study room to the map, but in order to display the room in the bookings table, the scripts themselves must be modified.

The Floor Buttons Script element of the Divi builder contains the script utilized to change floors. The changeFloor function is called every time the floor buttons are clicked and once when the website starts up. To get a new study room to show in this table associated with a specific floor, first navigate to the if statement corresponding to that floor. The writeUpcomingRow function handles all the logic for displaying future

availability. To add a new row to the table for a new study room, simply copy of the calls to this function and replace the 5-digit number with the room ID of the study room as it appears in the LibCal database.

```
if (value === '1') {  
    floorSelection.innerText = "Floor 1 Selected";  
  
    writeUpcomingRow(bookings, upcomingTable, 61355, next_hour);  
    writeUpcomingRow(bookings, upcomingTable, 36128, next_hour);  
}
```