Status Report 2

Group 6: RoomWrangler
Kate, Connor, Yumna, Rachel
7 April 2024

1. **Introduction**
    a. **Highlights**
        i. **What was the plan for this iterations?**

The overarching goal for this sprint was to make the product more practical by adding a search by time window feature, showing a calendar view of room availability, and allowing users to see and delete reservations they have made. We also planned to finalize our data model so that we can begin inputting real data to make the application of real use. Additionally, we planned to meet with customers to get updated requirements and feedback on our product. Here is our generalized sprint log:

1. Finalize data models
2. Add ability to search for a room by reservation time window and filter by room features
3. Develop ability to change users' roles
    a. In the future, each role will come with different permissions
4. Input data about rooms and standing reservations on one floor of Watson
5. Create a Calendar component
    a. Create functional calendar with no data based on design
    b. Get calendar to pull in Reservation data from in-app created reservations and any external data
6. Meet with stakeholders
    a. T&I
    b. Student ATs and non-AT students

        ii. **Highlight what the team accomplished**

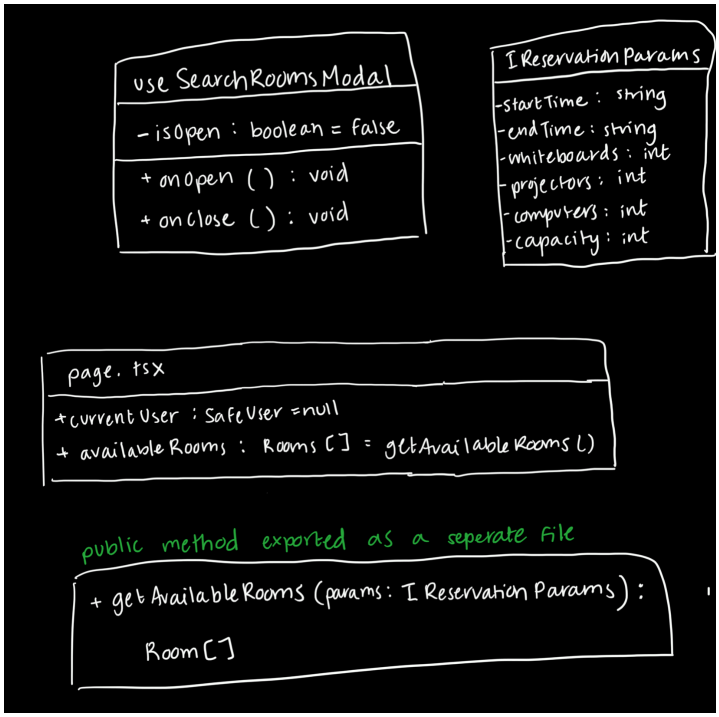
We completed all items on the sprint log. Our greatest accomplishment was the implementation of the search feature and the calendar functionality. We added the ability to search for a room by your desired time window. This was highly desired by customers. We also added optional search filters such as projectors and whiteboards. Another big accomplishment was the calendar view of room availability, which was mentioned by multiple customers, including Dr. Lim, T&I, and our AT contacts. In addition to the items on the sprint log, we began developing a "My Reservations" page which displays all reservations made by the user. This page is still a work in progress, but the foundation is set up for us to finish it in the next sprint.

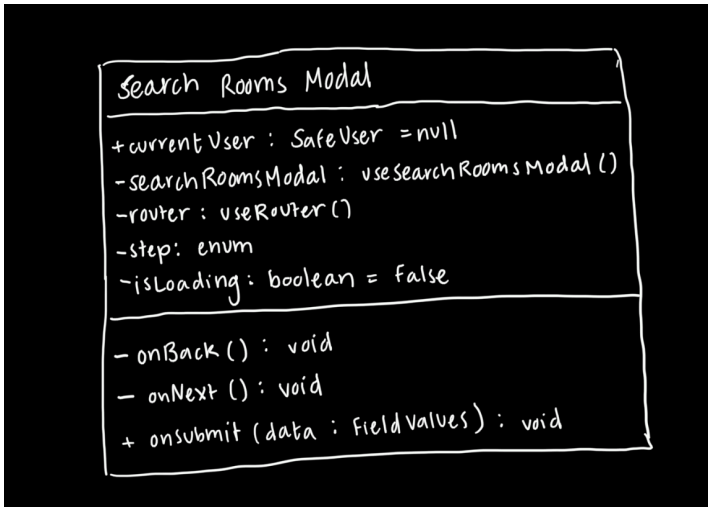
b. **Changes: Summarize any major changes since Status Report 1. Include each change's date, motivation, description, and implications. If there were none, not that there were no changes**
    i. Edit/Delete Reservation Ability (3/23): The initial product backlog included the ability to delete and edit reservations. While planning for this sprint, we considered changing this to only an ability to delete reservations. If a user needs to change their reservations, they could simply delete it and make a new one. We ran this idea by student stakeholders and they agreed. Since an edit reservation functionality would be high cost and low importance, we decided to remove this from our product backlog. Our product now will only include an ability for a user to delete a reservation, but users will not be able to edit a reservation
    ii. Data pulling (3/30): In Status Report 1, we had planned on pulling in external data about rooms and standing reservations from EMS. After meeting with T&I, we realized that accessing data within EMS will be very difficult. Additionally, even if we did have access, the data is not in an easily usable form for us. So, we decided that for the purposes of this project, we will input data manually. If this project were to continue and be put on the market, this may not be desirable. However, for our short term and small scope project that only includes one building, this will save us a lot of time and effort as opposed to figuring out how to access EMS and import the data.

2. **UML Class Diagram: Draw a class diagram that only includes the important classes your group mainly worked on during this iteration. Identify a single owner on the team for each class, even if multiple team members contribute.**

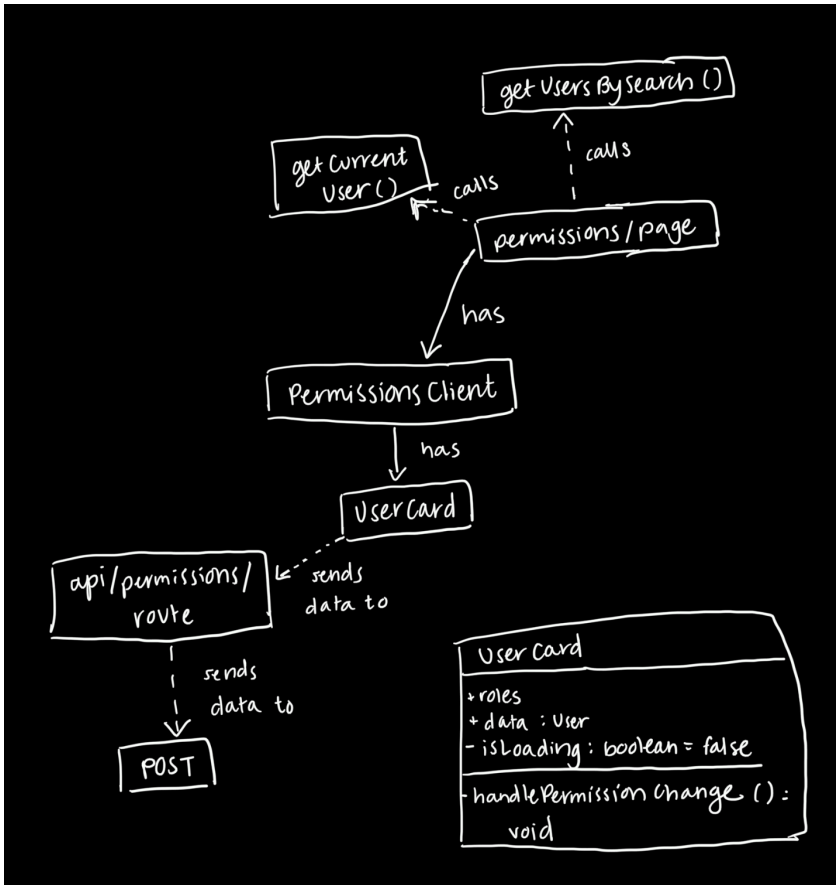
We modeled class relationships for the classes used to build the core functionalities we worked on during this sprint: search for a room, calendar view, permissions, and upcoming reservations. Class relationships are modeled in the first picture, and detailed diagrams for each class we made major modifications to are shown below the first picture. We are using Next.js for our project, which makes modeling the visibility of component attributes, along with the relationships between our components, difficult.

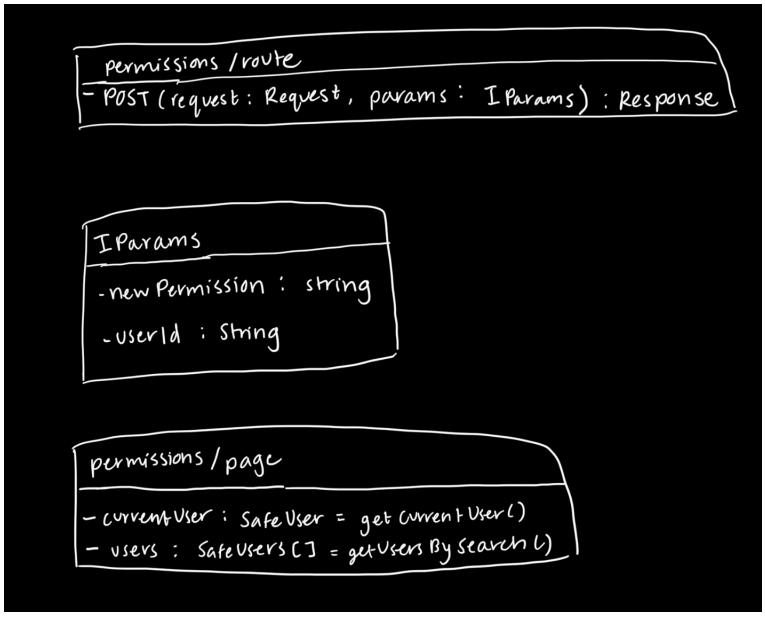
A. Classes used for Search Room functionality (Owner: Kate)

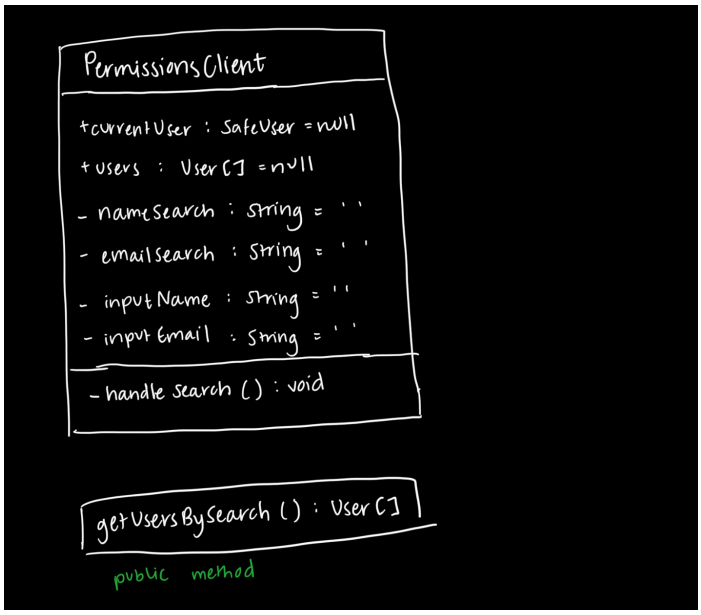




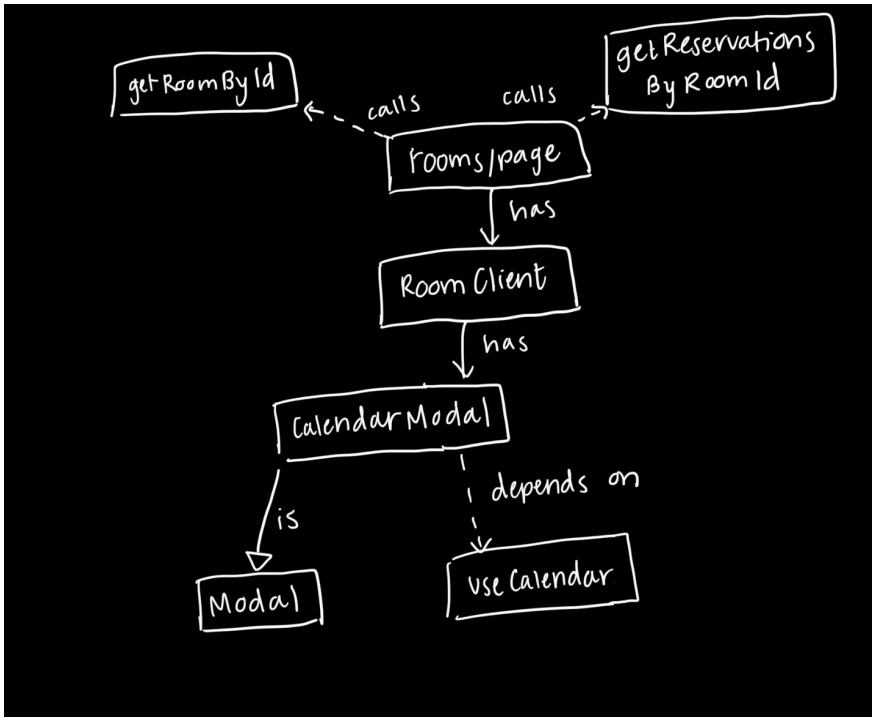
B. Classes used for Permissions functionality (Owner: Connor)

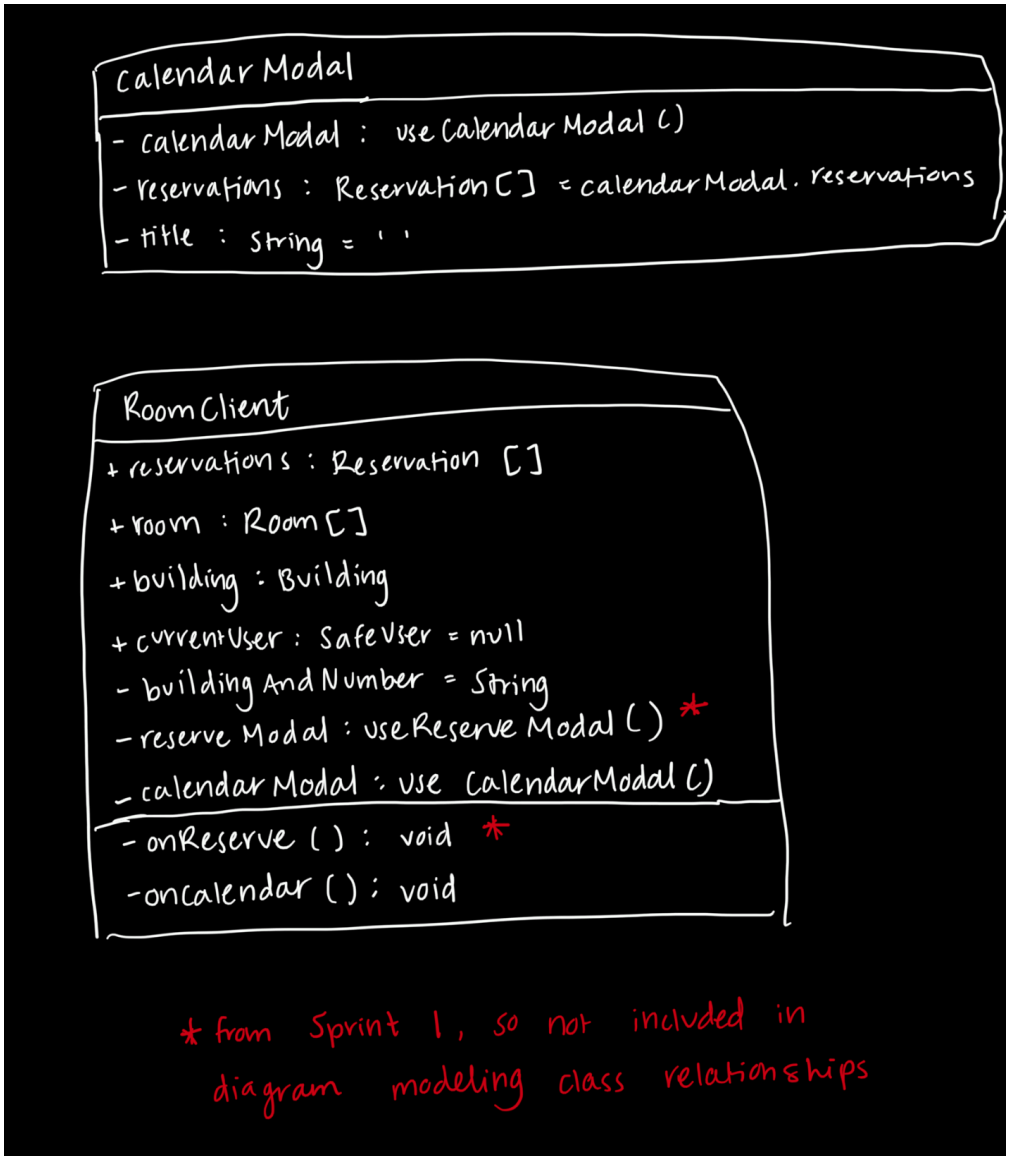


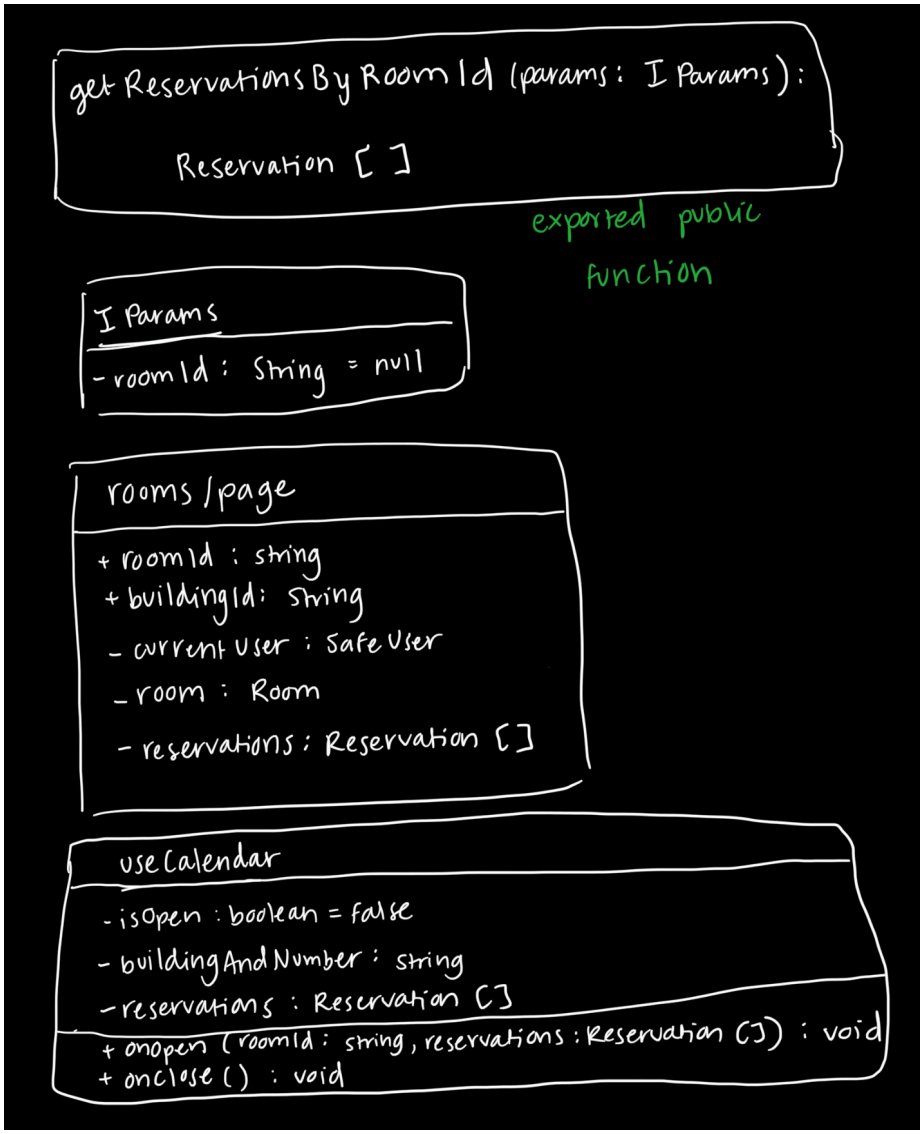

permissions / route
- POST (request : Request, params : IParams) : Response
IParams
- newPermission : string
- userId : String
permissions / page
- currentUser : SafeUser = getCurrentUser()
- users : SafeUsers[] = getUsersBySearch()


PermissionsClient
+currentUser : SafeUser = null
+users : User[] = null
- nameSearch : String = ''
- emailSearch : String = ''
- inputName : String = ''
- inputEmail : String = ''
- handleSearch() : void
getUsersBySearch() : User[]
public method

C. Classes used for Calendar functionality (Owner: Connor)



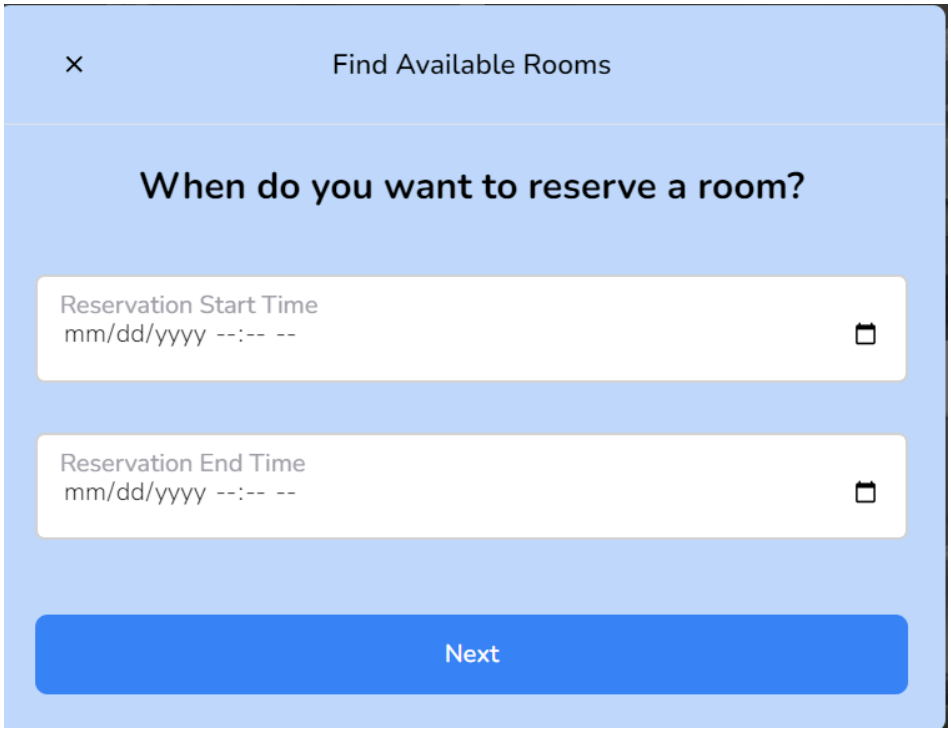

Calendar Modal
- calendarModal : useCalendarModal()
- reservations : Reservation[] = calendarModal.reservations
- title : String = ''
RoomClient
+ reservations : Reservation []
+ room : Room[]
+ building : Building
+ currentUser : SafeUser = null
- buildingAndNumber = String
- reserveModal : useReserveModal() *
- calendarModal : useCalendarModal()
- onReserve() : void *
-oncalendar() : void
* from Sprint 1, so not included in diagram modeling class relationships

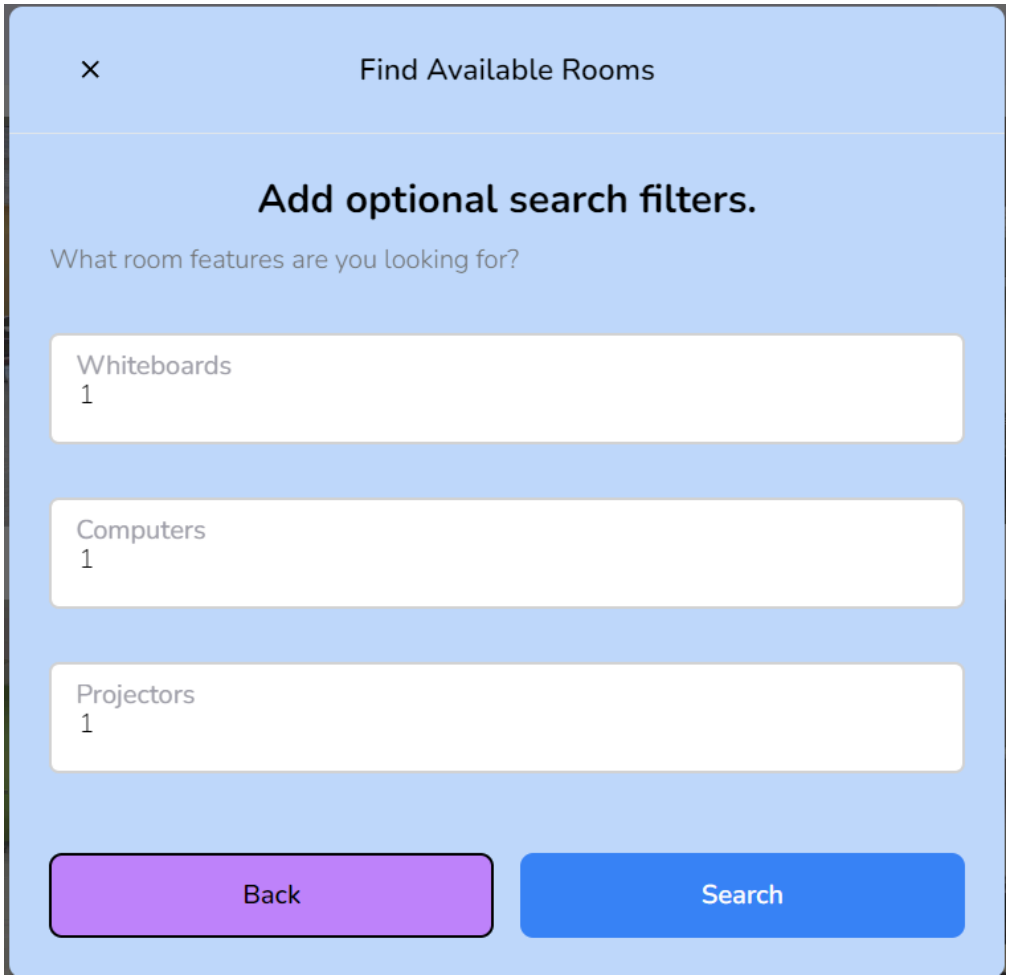


3. **Current Status: Add screenshots of the parts that are working. Map the screenshots to the class diagram. In other words, which class does the feature you show in the screenshot belong to? Multiple classes may belong to a single feature. Clearly describe what those are**

These screenshots are from working features that we modified during the current sprint. Features that we did not modify during the current sprint are not included in this section, as their implementation has not changed since Status Report 1.

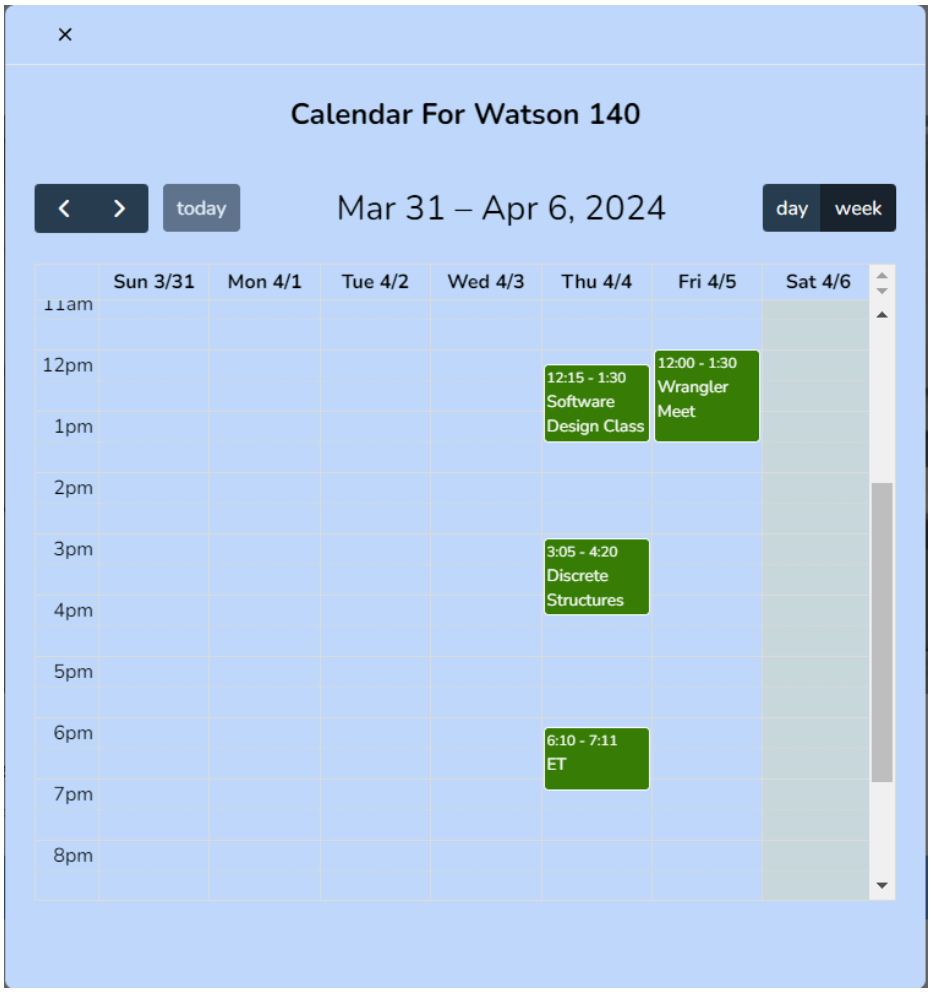
A. Search functionality


×
Find Available Rooms
When do you want to reserve a room?
Reservation Start Time
mm/dd/yyyy --:-- --
Reservation End Time
mm/dd/yyyy --:-- --
Next


×
Find Available Rooms
Add optional search filters.
What room features are you looking for?
Whiteboards
1
Computers
1
Projectors
1
Back
Search

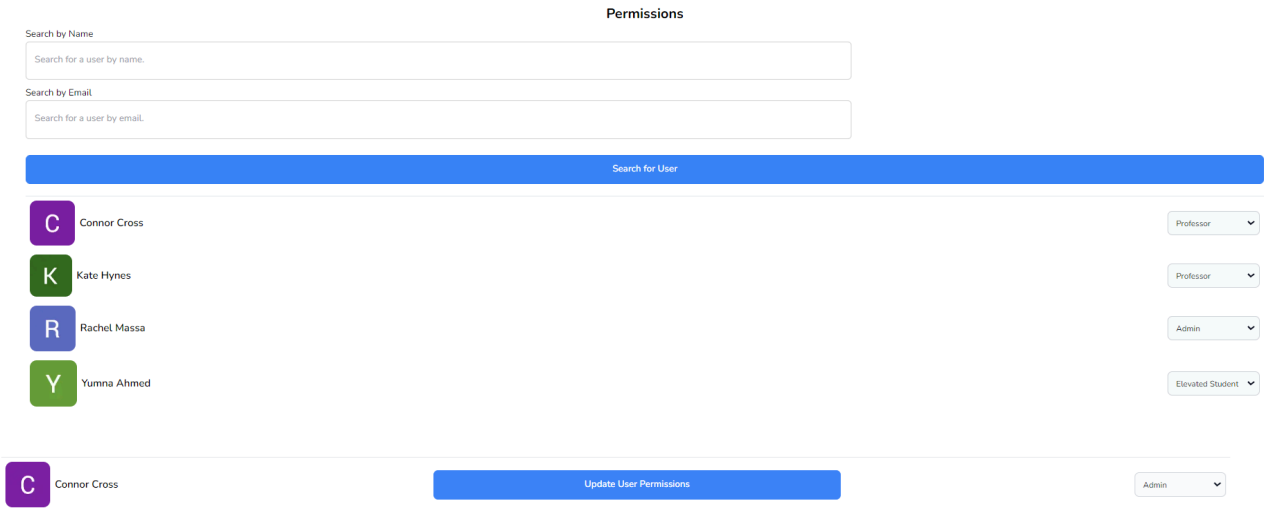
The UI shown in the screenshots is a part of **SearchRoomsModal**. We use the hook **useSearchRoomsModal()** to encapsulate logic related to SearchRoomsModal's UI. SearchRoomsModal inherits properties and behavior from our generic **Modal** class. When the user selects the Search button, the information from the modal is passed through url parameters to **page** (the home page), which triggers **getAvailableRooms()**.

B. Calendar



The UI shown in the screenshots is a part of **CalendarModal**. We use the hook **useCalendarModal()** to encapsulate logic related to CalendarModal's UI. CalendarModal inherits properties and behavior from our generic **Modal** class. **rooms/page** is the page that renders **RoomClient**, which contains CalendarModal. When the user selects the View Calendar button on the room page, it triggers **getReservationsByRoomId()**. Any reservations found by this action are passed into rooms/page and rendered on CalendarModal through RoomClient.

C. Permissions page



The UI shown in the screenshots is a part of **permissions/page**, which contains the **PermissionsClient** component. permissions/page calls **getUsersBySearch** and passes that data into PermissionsClient. PermissionsClient contains **UserCard** components for each user passed in. When the user updates a user's permission within UserCard, UserCard sends data to the route defined in **api/permissions/route**, which makes a **POST** request to the database.

4. **Project Management**
    a. **Continue to maintain the Change Log. Add any new changes to the project, tracking the date and description of each change. Use the table below:**

Date	Description
Old Changes	
2/13	Had initially wanted our system to work for all academic buildings. As we gained a clearer picture of what can be accomplished by four college students within a semester, we decided to limit the scope of our project to only one academic building, with potential for adding more if the project continues after the semester is over.
2/28	Decided we want ATs to schedule through us. Initially, we were not sure if we wanted our product to handle AT scheduling. When we met with the Registrar's Office, they told us that they would like this.
3/21	Decided to discuss possibilities for putting our product into production with T&I. Initially, we weren't sure if we wanted RoomWrangler to be pushed to production, but now we think we may have time to incorporate that into our plans and can pursue this aspect.

New Changes	
3/23	The initial product backlog included the ability to delete and edit reservations. After team discussion and meeting with users, we decided to take the edit reservation ability out of the product backlog and include only a delete reservation ability.
3/30	Had initially planned on importing room and reservation data from EMS directly into our database. After meeting with T&I and understanding more about EMS, we decided that importing data in this way is too time-consuming, and we will not be able to implement it. We decided to input this data manually for the purpose of this class project.

5. **Review and Retrospective**
    a. **What went well?**
        i. We were able to implement the features and functionalities that we set out for this sprint. We also had productive meetings with stakeholders including students, ATs, and T&I.
    b. **What didn't go well?**
        i. T&I is not sure that they will be able to implement our product at Davidson. They walked us through their requirements for software applications they use for the school. We certainly will not be able to meet all of their important requirements by the end of the semester, and as students we may not be able to even if we were to continue this project after the semester is over.
    c. **For the goals that were not met, what were the issues?**
        i. Adding in existing reservation data for Watson rooms.
            We realized that EMS has inconsistencies about reservation data for Watson specifically, and all requests go through the registrar. Hence, we were unable to get data directly or indirectly from EMS. We had to go to the registrar instead of EMS.
    d. **How do you plan to overcome the issues?**
        i. Adding in existing reservation data for Watson rooms.
            Yumna contacted the registrar and we now have the data. We will work on importing it to our site.
    e. **What do you plan to do differently in the next iteration?**
        i. We will have students test our product in a way that gives us quantifiable data.
6. **Team Management**
    a. **What were the team roles for this iteration?**
        i. Kate: Developer

ii. Connor: Developer
iii. Yumna: Scrum Master; Developer
iv. Rachel: Product Owner; Developer

b. What did each team member contribute?

i. Kate: Added Search for a Room modal, made adjustments to Rent Modal UI, drafted the sprint backlog
ii. Connor: Added Calendar view, Added Permissions page
iii. Yumna: Arranged meeting with T&I, determined best method for data collection, collected and imported data for Watson rooms
iv. Rachel: Arranged meeting with ATs, worked on Upcoming Reservations page and delete reservation functionality

c. What were the challenges regarding team management, e.g., regular meeting, etc.?

i. Scheduling meetings was difficult since we all have conflicting schedules, but we were able to overcome this challenge by having daily scrum style discussions before or after class and the meetings we held with stakeholders. Additionally, we communicated by text and shared Google Docs.

d. What are the plans to overcome the challenges?

i. We will continue to be flexible in our meeting and communication styles. As the semester has progressed, we have more frequently shared updates and asked questions on our group chat. This will continue throughout the next sprint.

e. If you were the third party who knows very well about your team, what suggestions would you give to your team?

i. Use ChatGPT to help with coding
ii. Ask each other for help when struggling with coding or debugging
iii. Start tasks early in the sprint

7. Goals for Next Iteration

a. Write the next iteration's product log.

i. Ability to search for a specific room
ii. Varied privileges for different types of users
iii. My Reservations Page
iv. Ability to delete reservations
v. Product holds all current room and reservation information for Watson

b. Write the next iteration's sprint log.

i. Add ability to search for a specific room
ii. Add a toast message indicating search successfully completed/not successful
iii. Determine what features should be available based on privileges

iv. Implement privileges & permissions system allowing certain privileges only to users who have a corresponding determined permission level (indicated by their user role)
v. Finish My Reservations page
    Show date and time of each reservation
vi. Create ability to delete reservations
vii. Finish importing room data for Watson
viii. Import reservation data for Watson
ix. Clean up UI
x. Meet with student / AT stakeholders to test our product and ensure they are satisfied

c. Other than the issues discussed in Section 5, i.e., Review and Retrospective, what potential challenges do you see in the next iteration?

i. My Reservations Page: Currently, the work-in-progress My Reservations Page may be set up to show ALL reservations rather than only future reservations. We will have to decide if we want all reservations to be shown or only future reservations. If we only want future reservations to be shown, we will have to figure out how to do that.
ii. Fake data: We had previously been using fake rooms and reservations in our database to test our product. Now that we have real rooms, we will have to deal with the old reservations corresponding to the fake rooms.
iii. Privileges / permissions: We have not yet finalized which different privileges we will allow for each user role. Although we know of some major restrictions we must have for regular student users, we may run into additional restrictions or necessary permissions for other user roles. As we determine each user role's permission level, we may end up having questions for the registrar.

d. Briefly explain how your team would overcome each challenge.

i. My Reservations Page: We will ask users if they would like to see only upcoming reservations or old ones as well. If they want to see both, we will probably want to separate them somehow. So, either way, we will need to code in some logic to get only new reservations (and possibly some logic to get only old reservations as well). The Reservations modal in our repository already contains similar logic, so we will look there for guidance.
ii. Fake data: We will assign someone in the group to go over the database and remove all reservations for rooms that no longer exist. Luckily, we did not make a lot of fake reservations, so there is no need to automate this process.

iii. Privileges / permissions: We will be sure to start early on this task so that we can address any questions or concerns that may arise. Additionally, when we do finalize privileges allowed for each different user role, we will document this information clearly on a shared Google Doc.