

1.

**Edit This Room**

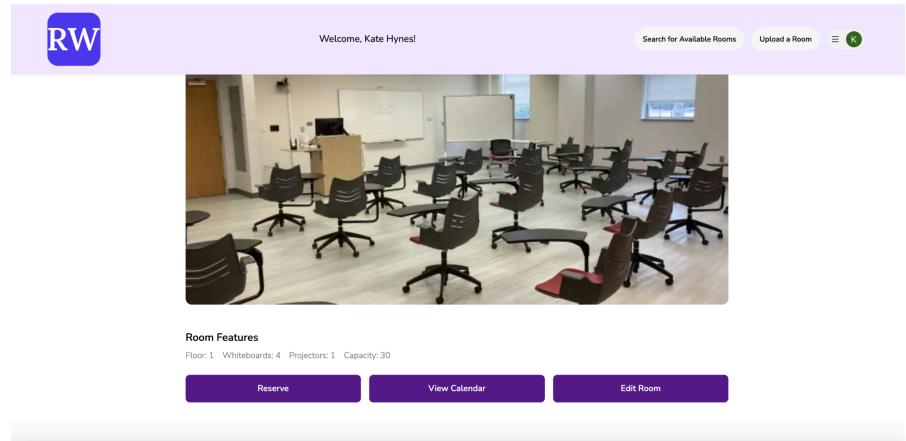
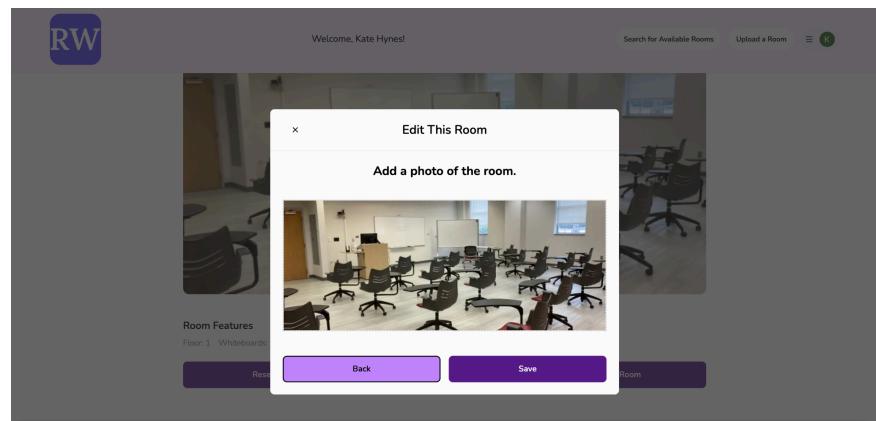
**Update Room Info**

Floor	1
Whiteboards	4
Computers	0
Projectors	1
Capacity	30

Room Features  
Floor: 1 Whiteboards

Reserve View Calendar Edit Room

Next

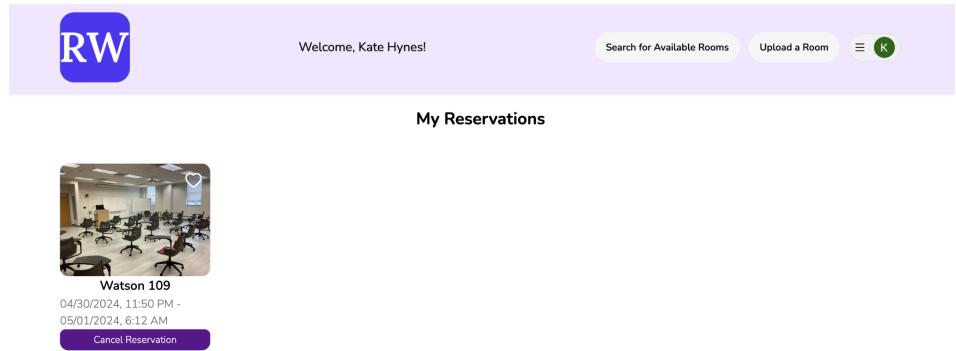


3.

- This sequence of screenshots shows the process of editing room information for rooms which have already been uploaded to the database. It can be found in the “Edit Room” box in the system description diagram. The inputs it takes are for changing the number of projectors, whiteboards, computers, capacity, and the image displayed for the room. It then sends these inputs to the database, and regenerates the output based on the updated room. This functionality is only available to admins. We thought it

would be important in case admins want to change room information after uploading a room.

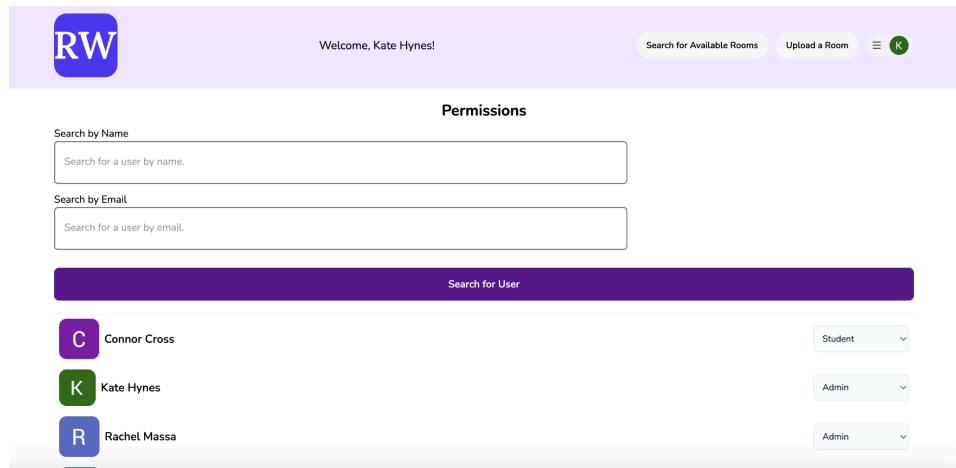
#### xi. Reservations Page



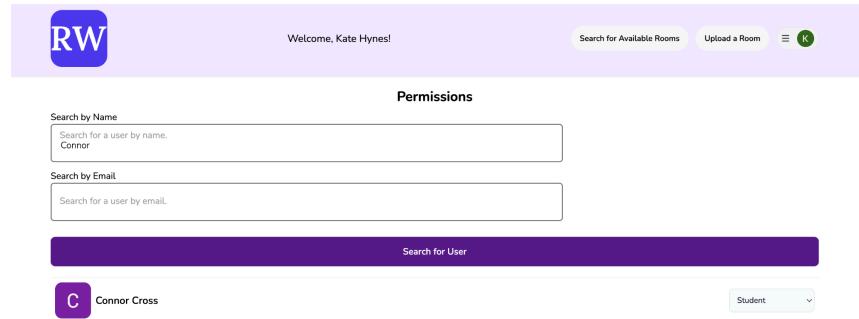
1.

- a. This screenshot shows the reservation page and the process of canceling a reservation. The reservation page is located in the “My Reservations” box in the system description diagram. The user can access this page through the user menu. It displays all their reservations based on the information stored in the database. The only input here would be the cancel button, which, upon clicking, wipes the reservation from the database and removes it from the page. The output would be all the reservations from the database. We thought this would be important in case someone accidentally makes a reservation or has to cancel last minute and wants to open up the space for someone else.

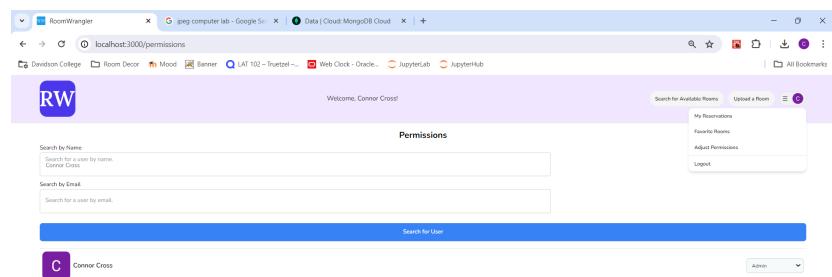
#### xii. Permissions



1.

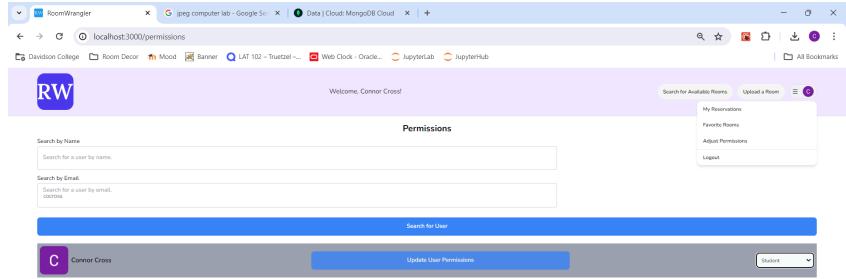


2.

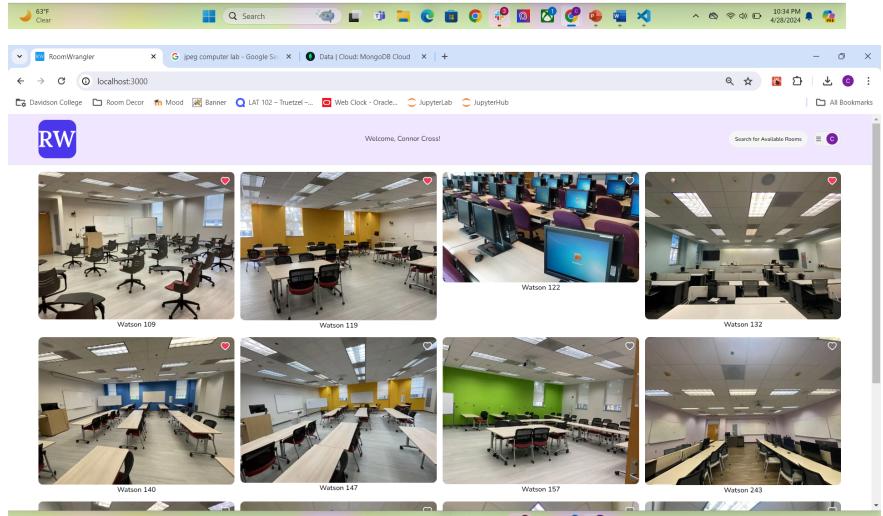


3.

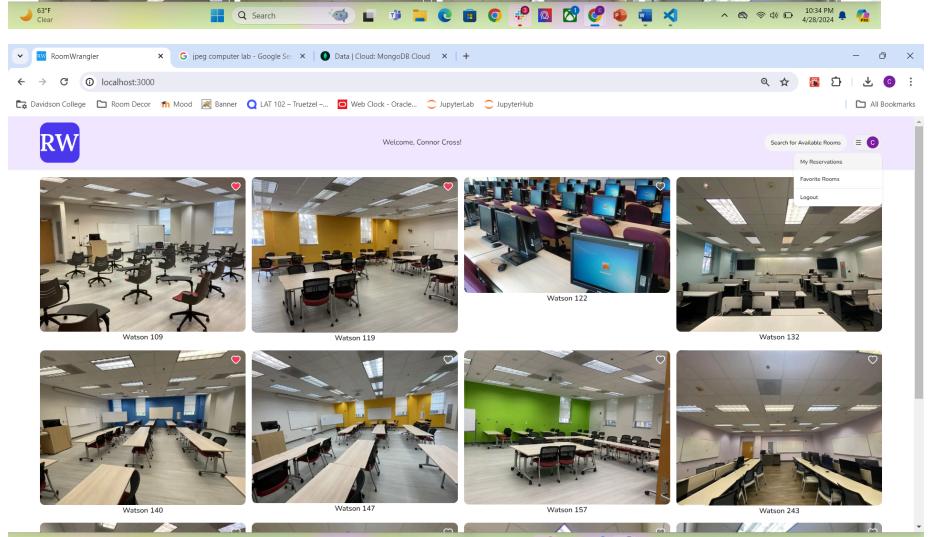




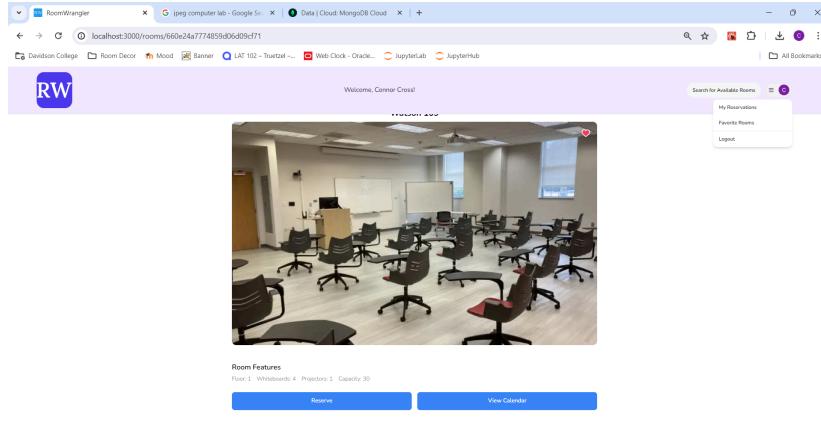
4.



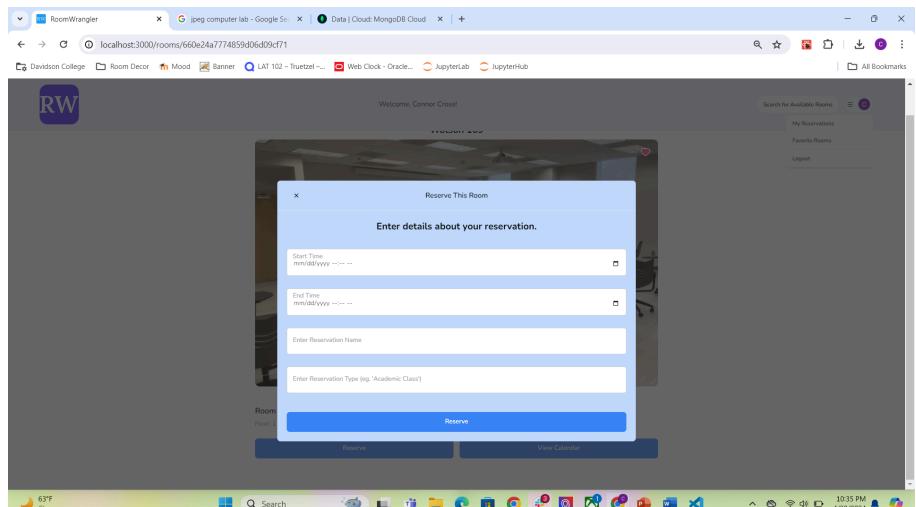
5.



6.



7.



8.

- a. All of these screenshots show how to change someone's permissions as well as what effects occur after permissions are changed. It can be found in the "Permissions" box on the system description diagram. Images *a-d* show the process of changing someone's permissions, as well as how to search for a specific user. Pictures *d-h* can be contrasted with other images above, and you will see the effects that changing the permissions had. The navbar no longer has the "Upload Room" option, the user can no longer edit rooms, the user menu no longer displays the permissions page option, the user can no longer make weekly reservations, and the user can no longer edit rooms. We thought this would be important so that students wouldn't have the ability to upload new rooms, change other people's permissions, edit rooms, or make weekly reservations.

- c. Software Testing  
i. Test Cases

*List five test cases to test five different features of your system. Then, justify why each case is important. The list may not have been changed since the group's last proposal (Homework 4). In this case, your group is welcome to copy and paste from the assignment.*

1. **Upload a Room:** When a user clicks the “Upload a Room” button, a modal will appear on the screen. The user should be able to close this modal and navigate between screens using the buttons at the bottom of the modal. When a user clicks the “Upload” button on the last screen of the modal, a room with the features the user specified in the modal should be created in the database. The user should see a toast message indicating that the room was successfully created.
  - a. This feature is important because it proves that our system can successfully (1) receive a command from clicking a button (Upload a Room), (2) open the correct modal, (3) display fields for and gather user input, and (4) update the database with new room information. It allows rooms to be added to the database as the campus changes over the years. Using this feature to add rooms is easier for a user than inputting information directly into the database.
2. **Reserve a Room:** When a user clicks the “Reserve” button, a modal will appear on the screen. If the user enters valid input in all fields on the modal, when the user clicks “Reserve” to submit the reservation request, a reservation record should be created with the information specified in the modal. A toast message should indicate that the reservation was successfully created.
  - a. This feature carries out technical functions similar to feature #1: receive command (Reserve), open modal, display fields/gather user input, and update database. Additionally, it proves that our system can (1) correctly handle date/time data, and (2) display Toast messages to the user, confirming that their action was successful so they do not have to blindly trust the system. This feature as a whole provides the core functionality of our system: allowing users to make reservations.
3. **Search for a Room:** When the user clicks the “Search For a Room” button, a modal should appear on the screen. The modal

should have a start time and end time input field. The modal should have optional input fields for desired room features. When the user clicks “Search,” if the user has entered valid input for all required fields on the modal, any rooms that match the search criteria and are available during that time frame should be displayed on the Home page.

- a. This feature carries out technical functions similar to feature #1 and #2: receive command (“Search For a Room”), open modal, and display fields/gather user input. It also proves that our system can successfully (1) search through database to gather rooms fitting search criteria, and (2) update Home Page display to show only rooms fitting search criteria. This cuts down user hassle in looking through rooms to find one that fits their needs.
4. **Room Information Page:** When a user clicks on a Room Card component displayed on the Home page, they should be navigated to the Room Information page for that room. This page will display the room’s building, number, features, and an image, along with the Reserve button.
  - a. This feature is important because it proves that our system can successfully (1) open the correct new page (when user clicks on a room), and (2) display room information from database. The page that this feature launches shows users a bigger view of the room and all of its features. It is also where users will have the ability to reserve the room.
5. **Schedule for Room:** When a user is looking at the room information page, there will also be a button to open a modal containing the schedule of the room. When the user clicks on the button, it should display a schedule with all of the reservations made for the room during that week.
  - a. This feature is important because it proves that our system can successfully (1) correctly open the schedule model, (2) load in all the reservations from a given room stored in the database, and (3) help to successfully inform a user which times are available.

ii. Combinatorial Testing

1. *Define all the factors for your system. For each factor, define all the values you might use in the combinatorial testing (Note: You do not have to create covering arrays. Just list the factors and values).*
  - a. User Permissions: Admin, Professor, Elevated Student, Student
  - b. Device Types: Desktop computers, tablet, smartphone
  - c. Operating Systems: macOS, Windows
  - d. Browsers: Google, Safari, Firefox
  - e. Login Modal – Inputs
    - i. Username: Davidson username, non-Davidson username, empty input
  - f. Search Rooms Modal – Inputs
    - i. Specific Room: Empty input, Input that matches database entry, Input that does not match database entry
    - ii. Reservation start time: Empty input, valid future date, past date, date before end time
    - iii. Reservation end time: Empty input, valid future date, past date, date before start time
    - iv. Whiteboards: Empty input, Non-number, negative number, valid number
    - v. Projectors: Empty input, Non-number, negative number, valid number
    - vi. Computers: Empty input, Non-number, negative number, valid number
  - g. Upload Room Modal – Inputs
    - i. Building: No selection, valid selection
    - ii. Number: Empty input, valid input
    - iii. Floor: Empty input, Non-number, negative number, valid number
    - iv. Whiteboards: Empty input, Non-number, negative number, valid number
    - v. Projectors: Empty input, Non-number, negative number, valid number
    - vi. Computers: Empty input, Non-number, negative number, valid number
    - vii. Capacity: Empty input, Non-number, negative number, valid number
  - h. Reserve Modal – Inputs

- i. Reservation start time: Empty input, valid future date, past date, date before end time
- ii. Reservation end time: Empty input, valid future date, past date, date before start time
- iii. Reserve weekly: Checked, unchecked
- i. Edit Room Modal – Inputs
  - i. Floor: Empty input, Non-number, negative number, valid number
  - ii. Whiteboards: Empty input, Non-number, negative number, valid number
  - iii. Projectors: Empty input, Non-number, negative number, valid number
  - iv. Computers: Empty input, Non-number, negative number, valid number
  - v. Capacity: Empty input, Non-number, negative number, valid number
- j. Permissions Page – Search – Inputs
  - i. Name: empty input, invalid input, valid input
  - ii. Email: empty input, invalid input, valid input
- k. Permissions Page – Adjust Permissions
  - i. Dropdown: Admin, Professor, Elevated Student, Student
- l. Favorites: Toggle favorite, toggle unfavorite

## **8. Project Management**

### a. Change Log

Date	Description
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.

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.

## 9. Review and Retrospective

### a. Sprint Review

- i. *Are there any additional customer needs?*
  1. Inclusion of all classrooms across all academic buildings
    - a. We decided near the beginning of our project that this would be out of our scope
- ii. *What are some of the customer(s) requests that you could not accomplish?*
  1. Include all classrooms across all academic buildings
  2. T&I technology requirements
    - a. Software would not have to be maintained by Davidson
    - b. Security requirements
    - c. Importing data directly from EMS
- iii. *Any comments and feedback from the customer(s)?*
  1. Users reported that they like this system much better than EMS and they would use it if it were deployed
  2. Users specifically liked the search functionality and the calendar view of room availability

### b. Sprint Retrospective

- i. *What went well?*
  1. We accomplished all of the must-do items on our sprint backlog and many of the optional items
  2. Our product would be ready to deploy at this stage
  3. We made our product more user friendly
- ii. *What didn't go well?*
  1. We had to edit some of our fields to allow for new changes during this sprint
  2. We thought we may surpass the limits of our free database usage. Luckily we did not.

- iii. For the goals that were not met, what were the issues?*
  - 1. We met all necessary goals for this sprint
  - 2. We were not able to meet all low priority goals, such as sending automatic email confirmations for reservations and adding other users to reservations. These items would have been relatively high cost as they are not like other work items we have completed and would require incorporating Outlook. We did not have time in our project to incorporate these low-priority requests.
- iv. How you could have done differently?*
  - 1. If we really wanted to incorporate all goals, we could have tried to plan better from the beginning, putting more tasks into each sprint. However, we did dedicate a lot of time to this project during each sprint as it was. Adding more tasks would likely have resulted in an unreasonable amount of time spent.

## 10. **Team Management**

- a. What were the team roles?*
  - i. Connor: Developer
  - ii. Kate: Developer
  - iii. Yumna: Scrum Master; Developer
  - iv. Rachel: Product Owner; Developer
- b. What did each team member contribute?*

Besides working on tasks below, each member of the team also contributed to tasks such as documentation efforts, sprint plannings and task prioritization, attending meetings with stakeholders/ customers, preparing status reports and presentations, and watching tutorial videos helpful to the project.

  - i. Connor: Set up coding environment, integrated APIs, upload room, navbar, calendar view, permissions page, favorites page
  - ii. Kate: Set up database, search modal, reserve modal, edit room modal, favoriting functionality
  - iii. Yumna: Drafted sprint backlogs, conducted testing, determined best method for data collection, collected and imported data for Watson rooms, adding to the database and data model
  - iv. Rachel: Handled meetings with customers, kept product backlog, conducted testing, developed “My Reservations” page
- c. What were the challenges regarding team management, e.g., regular meeting, etc.?*

- i. In the beginning, we did not specifically designate certain tasks to certain people or come up with deadlines for any tasks. This resulted in some uncertainties about generally who was doing what and when. Work tasks also ended up unevenly distributed. After the first sprint, we were able to plan better and avoid these issues. Scheduling meetings was difficult since we all have conflicting schedules.
- d. *What are the plans to overcome the challenges?*
  - i. After the first sprint, we handled sprint planning better to avoid issues of uncertainty and uneven work distribution. We assigned tasks to team members during sprint planning and also designated internal soft deadlines for tasks to be completed. We were able to overcome the scheduling challenge by having daily scrum style discussions before or after class and the meetings we held with stakeholders. We also held some meetings over Zoom. Additionally, we communicated by text and shared Google Docs. If we were to continue this project, we would use Zoom as our primary means of meeting, and we would text more frequently with updates and clarifying questions to make sure everything is running on track.
- e. *If you were the third party who knows very well about your team, what suggestions would you give to your team?*
  - i. Consider each person's areas of experience and knowledge when sprint planning
  - ii. Use ChatGPT to help with coding/ debugging
  - iii. Ask each other for help when struggling with coding or debugging
  - iv. Start tasks early in the sprint