

Final Report

Group 6: Connor Cross, Kate Hynes, Yumna Ahmed, Rachel Massa

RoomWrangler

30 April 2024

1. Introduction

Describe what the project is about, answering the following questions: • What is the title of your project? • What is the goal of the project? • What is the motivation for this project? • Who are the customers/users? • What development process/method did you use for your project (e.g., Agile-Scrum, Waterfall, etc.)? *Introduction is where you write an overview for the readers to get an idea of what to expect in the remainder of the report. You should not add too many details, e.g., algorithms or tools you used.*

- a. Through our project, RoomWrangler, we aimed to create a platform enabling users to browse and reserve classrooms for studying and other needs. We noticed that at Davidson, finding an open classroom for study group meetings and such can be difficult. Reserving a space ahead of time can also be difficult and is not ideal for college students and staff with ever-changing, busy schedules. RoomWrangler will serve Davidson college students, ATs, tutors, professors, and other staff members. Developing RoomWrangler, our team used Scrum methodologies. We held slightly longer sprints – we considered the time in between each status report to be one sprint.

2. Novelty

- a. *Describe about the novelty about the project. For example, what's new about the project compared to existing Software? Having a fancy UI is NOT a novelty. A novelty in research/project refers to introducing a new idea or a unique perspective that adds to the existing knowledge in a particular field of study. It involves bringing something fresh and original to the table that has not been done before or exploring an existing topic in a new and innovative way. Who are your potential competitors, i.e., existing software with a goal similar to or similar to your project? Explain what problems you plan to solve or your project aims to solve. Explain why your solution is better than them. Again, simply having a fancy button is not a novelty.*
- b. RoomWrangler enables users to browse and reserve rooms in ways that are unlike existing software.

- c. Currently, Davidson's main room reservation software is Event Management System (EMS). Under this system, Davidson users can request to reserve certain rooms across campus for events. Requests require detailed information and must be submitted at least two days prior to the event. Then, someone in the Registrar's Office approves (or denies, although this is uncommon) the reservation. The system is focused on the need to reserve rooms for events rather than study sessions, project group meetings, and other customer needs. Additionally, classrooms in Watson are not available to be reserved at all.
- d. Our system solves problems and avoids difficulties present in EMS. RoomWrangler is the first general-use reservation software for Watson classrooms at Davidson. Previously, students and professors would not know what rooms were available when in Watson, and they had no way of reserving the rooms. Additionally, with our system, users do not have to request spaces and wait for approval. Reservations are made instantaneously. This allows users flexibility in their decisions for room reservations.
- e. Another system used by Davidson that is more similar to RoomWrangler is LibCal. This is a system specifically for the Davidson College library that allows students to reserve one of four study spaces. Unlike EMS, this focuses on the need to reserve study spaces rather than the need to reserve spaces for events. RoomWrangler encompasses both of these needs and everything in between, allowing users to reserve classrooms for any reason. The study spaces available through LibCal are not full classrooms; their max capacity is four people. So, they are also not well suited for bigger groups. They also do not have the resources of a classroom that users may need such as projectors and whiteboards. Additionally, reservations through LibCal cannot be made more than a week in advance and can only last up to three hours. RoomWrangler allows users to make reservations for full classrooms as early or late as they like and does not limit reservation duration.

3. Customer

- a. Primary, Secondary, and Any Other Stakeholders
 - i. *Who is the primary customer outside the team?*
 - 1. Davidson users including students, ATs, ETs, professors, and other staff
 - ii. *Who are the secondary stakeholders?*
 - 1. Registrar
 - iii. *What do the stakeholders want?*
 - 1. Users want a centralized system for reserving an academic room without having to wait on external approvals. They want to be able to make and change their reservations easily and instantaneously. Additionally, they want to be able to see what rooms are available at any given time without having to physically check rooms.

2. The Registrar wants a lighter burden when it comes to scheduling.

iv. *Why?*

1. Students like to use classrooms to study and meet up for other purposes. In the college lifestyle, students often do not have plans made ahead of time. Additionally, their plans often shift. Therefore, they want to be able to reserve classrooms at their leisure when the need arises. Students often decide last minute that they want to study in a classroom. Currently, they do not know which classrooms are occupied and/or reserved without physically checking. People hate having to go door to door, knocking and peeking in to see if a room is occupied.
2. ATs and ETs sometimes have to adjust their schedules, so they want the flexibility to do so without having to haggle with the registrar to get approval.
3. The registrar currently handles all ET and AT scheduling directly. They are also responsible for approving all requests that come through EMS. They want to have more time and energy to devote to other tasks rather than this kind of scheduling.

b. Problems and Requirements

- i. Meeting Log: *Using the table below, write down all the dates you met with your customer(s), the problems the customer wants to solve, and the requirements the customer requested.*

Date	Customer	Description
2/28	Registrar's Office: Marc Jacobsen	Mr. Jacobsen, representing the registrar's office, wanted to solve the problem of his office having to handle a lot of hectic scheduling. He requested a system that would allow ATs and ETs to schedule sessions themselves rather than going through his office. He also wanted the system to be able to handle same-day reservations.
3/15	Students: Briana Harmon, Ella West, Jordan Miller	These students requested a way to reserve rooms for fun, informal happenings such as movie nights rather than just studying or hosting official events. They wanted a general-use reservation system. Additionally, they wanted to be able to see rooms that would be available during the time they would want to use one. They also wanted the option to instead see if a specific room would be available during the time they would want to use it.

3/25	T&I: Matt Jackson	<p>Mr. Jackson, representing Davidson T&I, wanted a software system to replace EMS. He wanted the system to utilize different permission levels for different types of users. Under this system, he wanted only the admin(s) to be able to see who made any reservation. He did not want users other than admin to be able to see this due to security risks. He also wanted a calendar component showing standing reservations. He did also have some tech requirements that we were not able to include in the scope of our project this semester. These requirements included heightened security measures, launching the platform as an app, importing data directly from EMS, and ensuring that the software would not have to be maintained by Davidson itself, rather use SaaS and have a third party oversee maintenance issues.</p>
3/26	Professor: Dr. Lim	<p>Dr. Lim, as a professor, wanted to easily find available rooms whenever the need arises. He wanted to see a calendar of availability for each room. He did not want much information beyond availability to be shown on this calendar so as not to make it look messy. He also wanted to see features of a room, like projectors or white boards. These could also be included as search filters, as recommended. However, he did not want us to include too many search filters and make the search process harder for users. He also wanted to be able to see reservations that he has made using the system.</p>
4/2	ATs: Elliot Jones and Penelope Hobbe Student: Briana Harmon	<p>These students, including ATs, wanted to be able to reserve rooms for study sessions and sessions they host using something better than EMS. They wanted to see a calendar of room availability like is shown on LibCal. They wanted to be able to see room information to choose a room best suited for their needs. They also wanted to be able to see the reservations they made. They wanted to see these in a block-view, with the nearest upcoming reservation shown first, followed by the next, and so on. They also wanted a way to look at their past reservations in case they needed to reference them.</p>

- c. Product Backlog: *List all the product backlogs for the project, i.e., a combined list of product backlogs from Sprints 1, 2, and 3, etc. For example, as a user of an*

online store xyz, I want to be able to search all the available products with keywords.

- i. As a student, I want to easily reserve a classroom for study sessions or other needs.
 - ii. As a professor, I want to quickly find an available room with desired features such as white boards and/or a projector whenever the need arises.
 - iii. As a student, I want to be able to reserve classrooms in the Watson building.
 - iv. As a room reserver, I want to choose a room based on its resources such as white boards and projectors.
 - v. As a room reserver, I want to easily browse rooms that are available during my desired time window.
 - vi. As a (student, professor, or other staff member), I want to be able to see all classrooms that are available at any given time without having to physically check them.
 - vii. As a room reserver, I want to be able to see the reservations I have made.
 - viii. As a room reserver, I want to be able to cancel a reservation I made if needed.
 - ix. As an administrator in the Registrar's Office considering implementing RoomWrangler for the school, I want to be able to upload a new classroom to the system.
 - x. As an administrator in the Registrar's Office considering implementing RoomWrangler for the school, I want certain features to be exclusive to administrators, professors, or ATs and ETs.
 - xi. As an administrator in the Registrar's Office considering implementing RoomWrangler for the school, I want to ensure that only users with a valid Davidson email address can utilize the software.
 - xii. As an administrator in the Registrar's Office considering implementing RoomWrangler for the school, I want to be able to edit a room's features (whiteboards, projectors, capacity, computers, floor) if they change or if I made a mistake when originally uploading the room.
- d. Overall Customer Experience: *Write a paragraph or two about the overall experience the user wanted to have from using your system (explicitly discussed with your group). Then, explain whether your current system satisfies their overall experience. Please be explicit about how you know your project is satisfied or dissatisfied.*
- i. Customers told us that they wanted a general-use reservation system and availability calendar for Davidson classrooms. Computer science professors specifically told us that they wanted a system for Watson. All users who had experience with EMS wanted a better alternative for

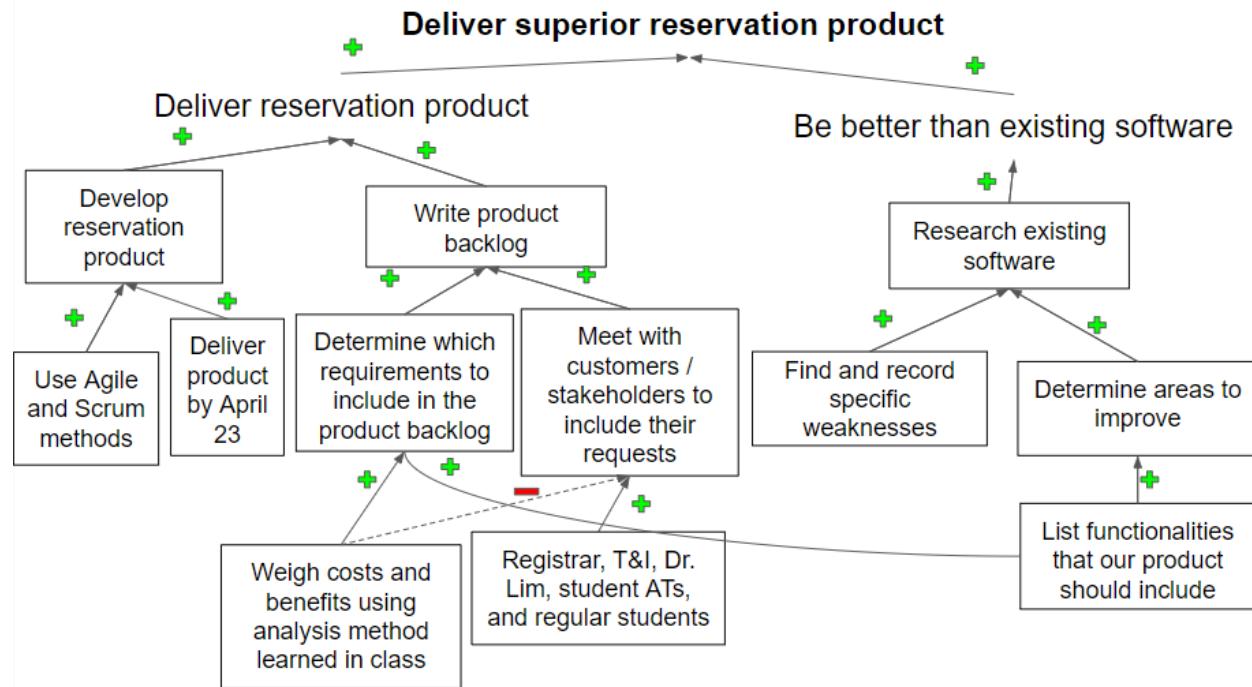
reserving rooms. ATs did not want to have to go through the registrar for room reservations. They wanted to be able to easily make a different reservation if they needed to. The Registrar also wanted an alternative to being responsible for scheduling and reserving rooms for AT sessions. General students wanted to be able to plan a study session in a classroom day-of, rather than having to plan at least two days in advance. They also wanted to be able to reserve classrooms for other things, such as movie nights. Aside from reserving rooms, users wanted to be able to easily find available classrooms that suit their needs whenever the need arises.

Professors and other users wanted an alternative to physically checking classrooms to see if they were occupied, and then hoping that no one would come in while they were using a room. As for finding a room that best suits their needs, users told us that they would like to see features of a room such as computers, projectors, and whiteboards. Overall, users wanted a system for Davidson classrooms that enables users to make reservations immediately and for any reason, shows room availability and features, and allows users to find a room best suited for them using a search feature.

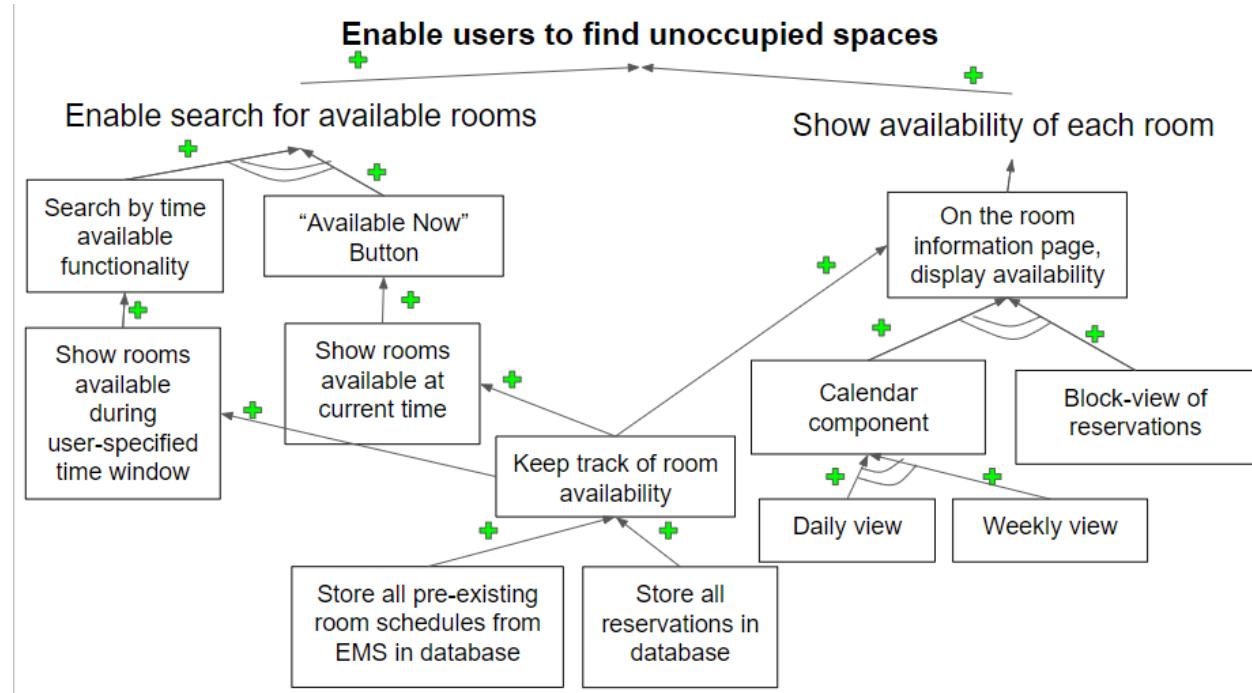
- ii. Our system satisfies user desires in nearly all aspects. We created a system that enables users to make classroom reservations instantaneously, without having to go through the Registrar. This satisfies all users' needs. On RoomWrangler, the information page for each room lists room features: number of computers, number of whiteboards, number of projectors, and capacity. We also included pictures of the rooms here and on the home page to better enable users to find the best room for them. On each room information page, RoomWrangler also displays a calendar view of that room's availability. We implemented a search functionality allowing users to search for: rooms available at a specific time; rooms with specific features; and/or a specific room. If a user needs a classroom suddenly, they can use this feature to find one that is available at that time. The only dissatisfying part of our system is that it only includes classrooms from Watson rather than all academic buildings. However, this does fit the needs of computer science professors we talked with.

4. SMART Goals

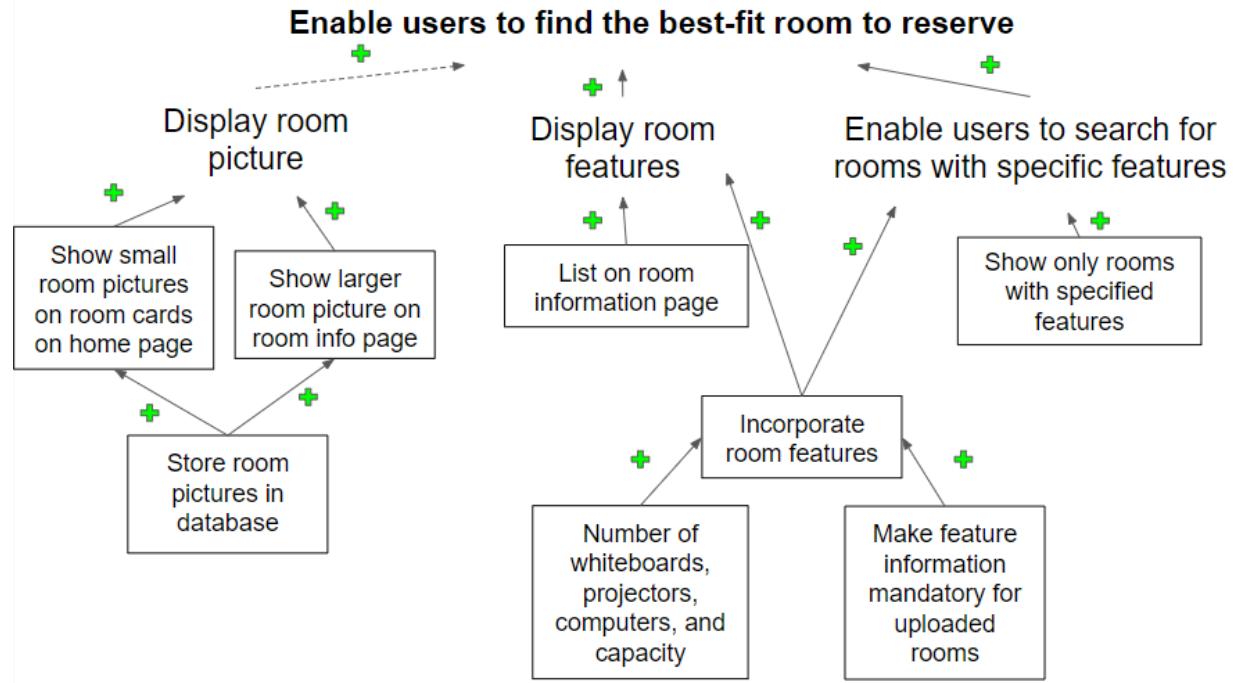
- a. *Select the three main goals of the project. Draw a SMART (Specific, Measurable, Achievable, Relevant, and Time-Bound) goal hierarchy for each goal. Make sure to follow the style and rules for the SMART goal hierarchy.*
- b. Goals: Deliver superior reservation product, Enable users to find unoccupied spaces, Enable users to find the best-fit room to reserve
- c. Goal: Deliver superior reservation product



d. Goal: Enable users to find unoccupied spaces



e. Goal: Enable users to find the best-fit room to reserve



5. Sprint Backlog

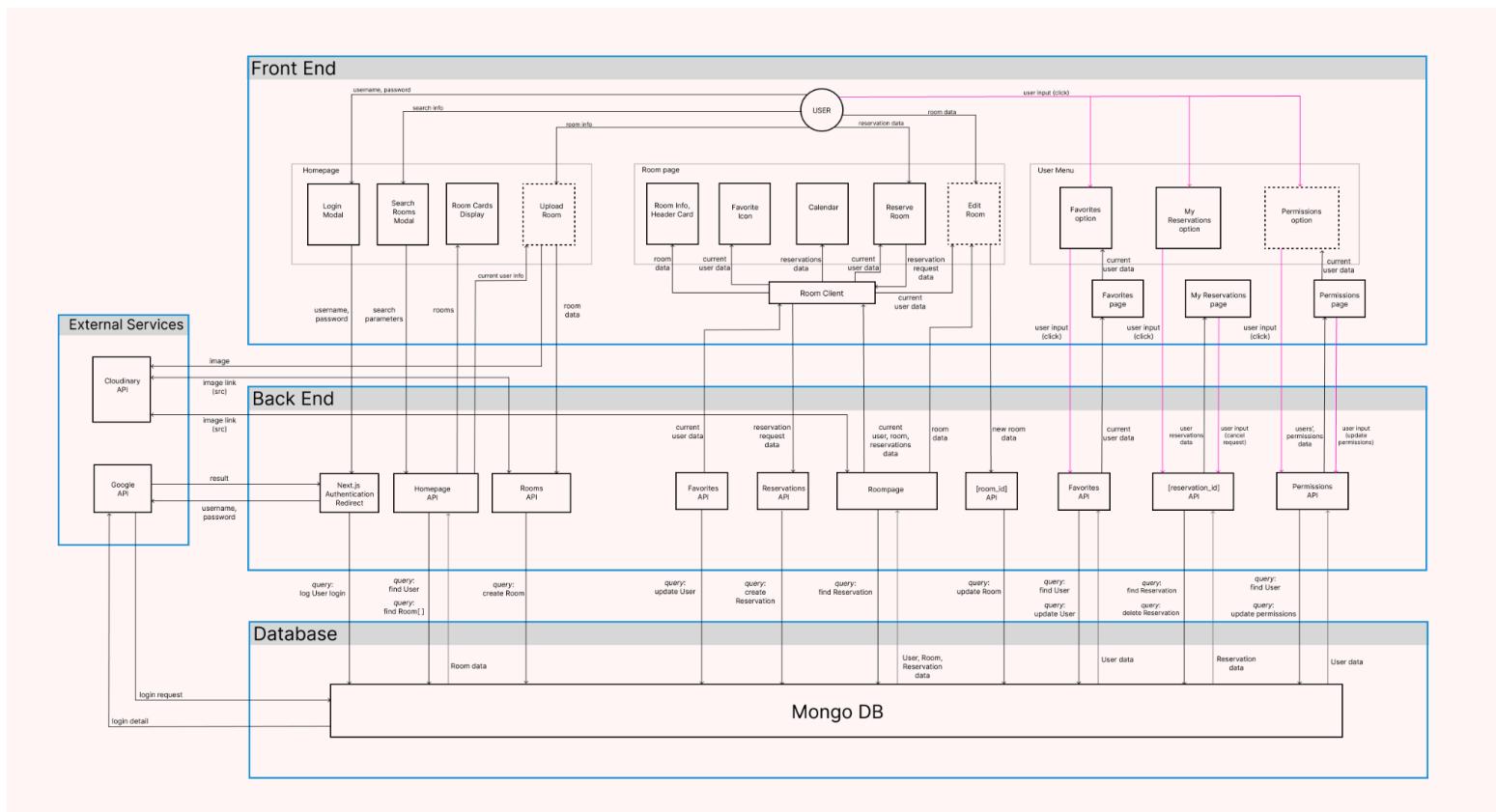
- a. List all the sprint backlogs for the project, i.e., a combined list of sprint backlogs from Sprints 1, 2, and 3, etc.
- i. Setup GitHub repo
 - ii. Setup coding environment
 - iii. Setup database environment
 - iv. Create login functionality
 - v. Ability to upload rooms to database and see them displayed on webpage
 - vi. Create basic workflow for reserving a room
 - vii. Finalize data models
 - viii. Add ability to search for a room by reservation time window and filter by room features
 - ix. Develop ability to change users' roles (so that different roles can be assigned different permissions)
 - x. Input data about rooms and standing reservations on one floor of Watson
 - xi. Create a Calendar component
 1. Create functional calendar with no data based on design
 2. Get calendar to pull in Reservation data from in-app created reservations and any external data
 3. Show each room's calendar on its room information page
 - xii. Gather and import room data on rooms in Watson
 - xiii. Gather and import existing reservation data on rooms in Watson

- xiv. Develop “My Reservations” page for users to view their reservations
- xv. Enable users to delete reservations they have made
- xvi. Add ability to search for a specific room
- xvii. Make a toast message appear for successful searches
- xviii. Change “My Reservations” page to only show upcoming reservations
- xix. Create a place for users to see past reservations
- xx. Implement different privileges for different user roles
- xi. Enable users with admin role to edit rooms
- xxii. Enable Admin, Professor, and Elevated Student users to make weekly recurring reservations

6. System Description

a. Block Diagram:

Figma Link: [RW System Diagram](#)



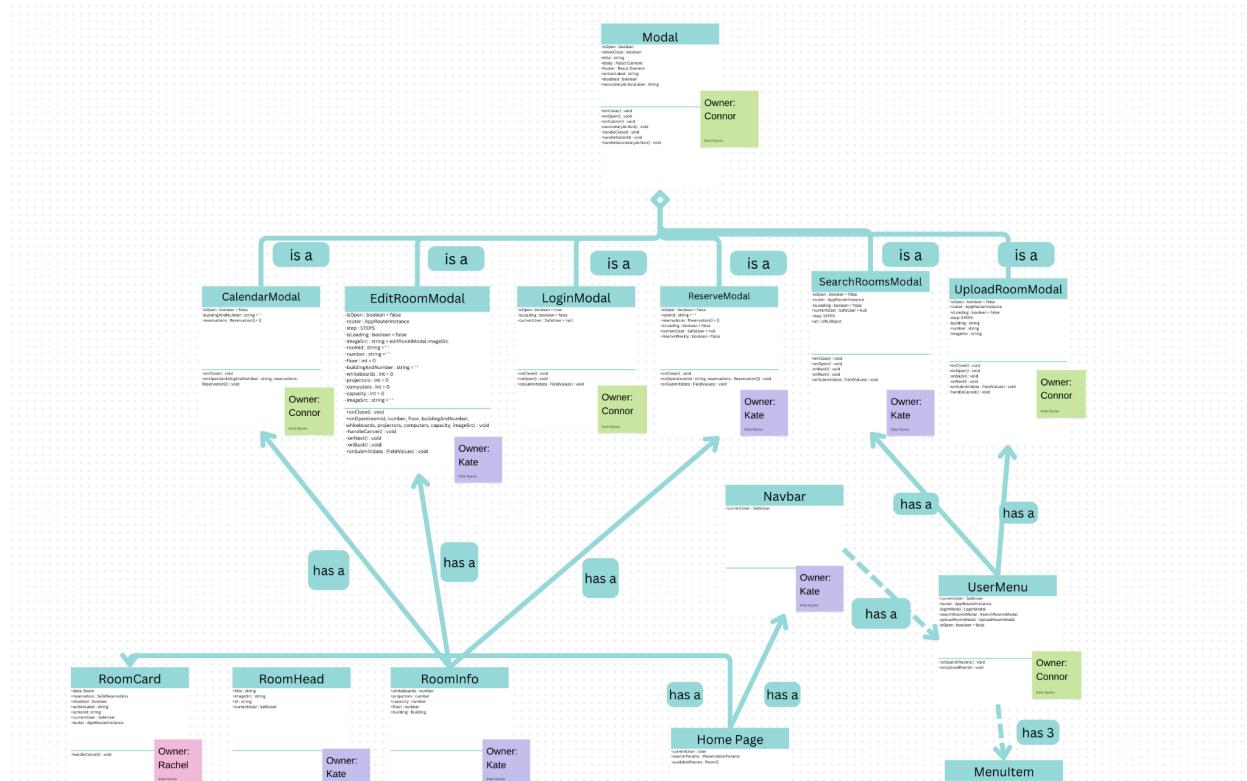
b. UML: Class Diagram

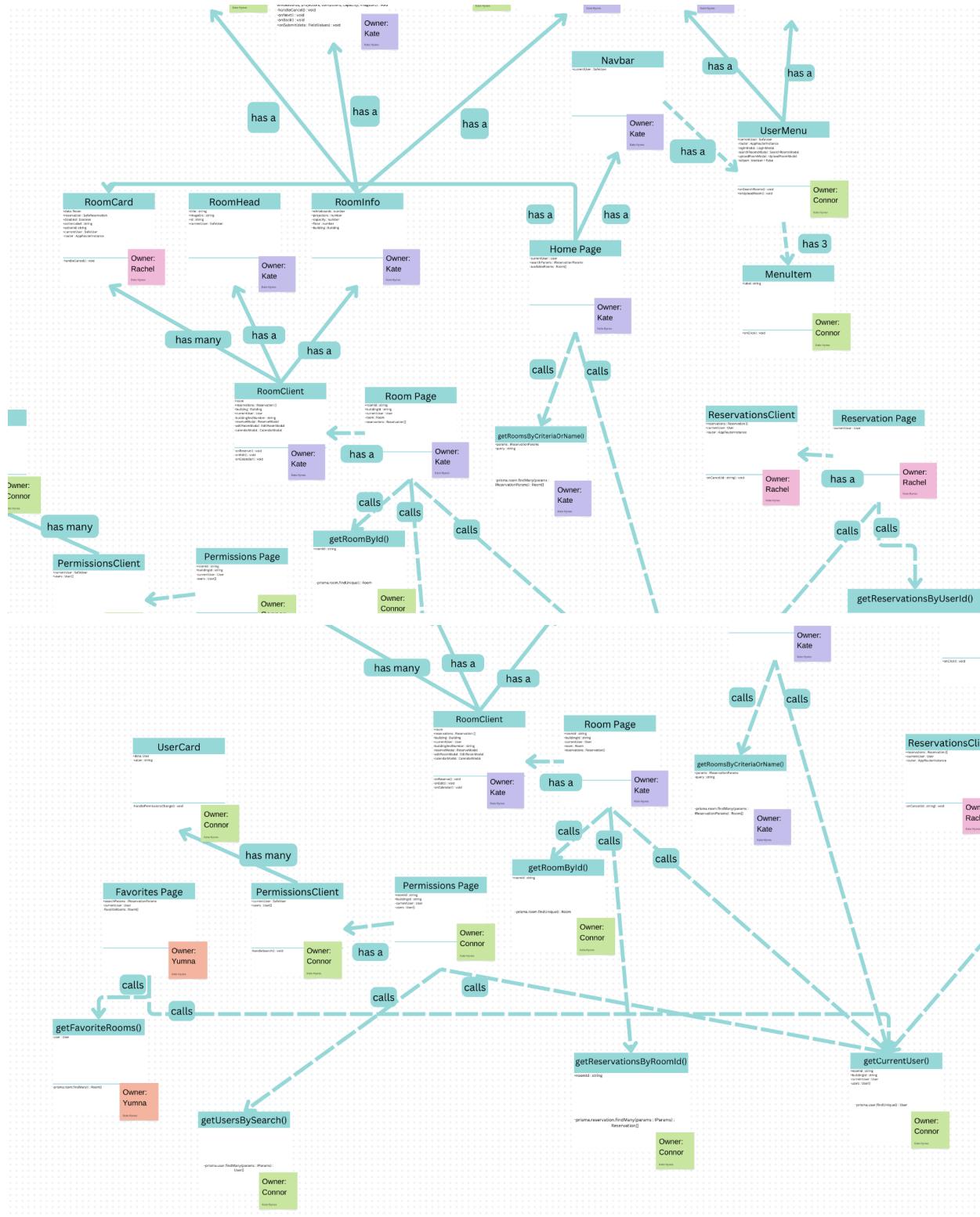
i. Draw a class diagram that includes all the classes in your system. Identify a single owner on the team for each class, even if multiple team members contribute. Make sure to use appropriate edges to show the relationship between the classes, i.e., dependency, generalization, or association. Use verbs for names style for the edges for association. Note that a UML class diagram is a developer's view of the significant classes in a design. We can highlight what we consider important about a class, while other information about a class can be omitted from the diagram.

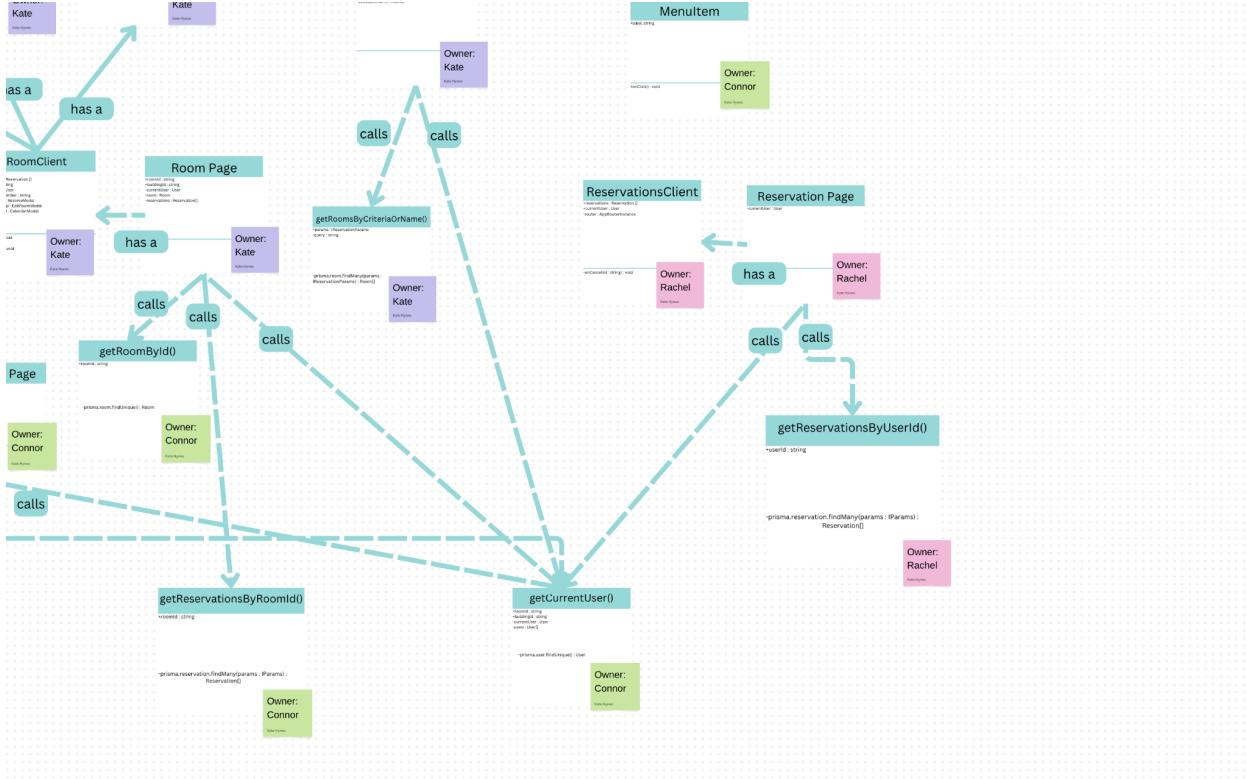
c. Link to Canva diagram:

<https://www.canva.com/design/DAGDukiX1zM/egEWleQ6NKb4xKBcFhN-7w/edit>

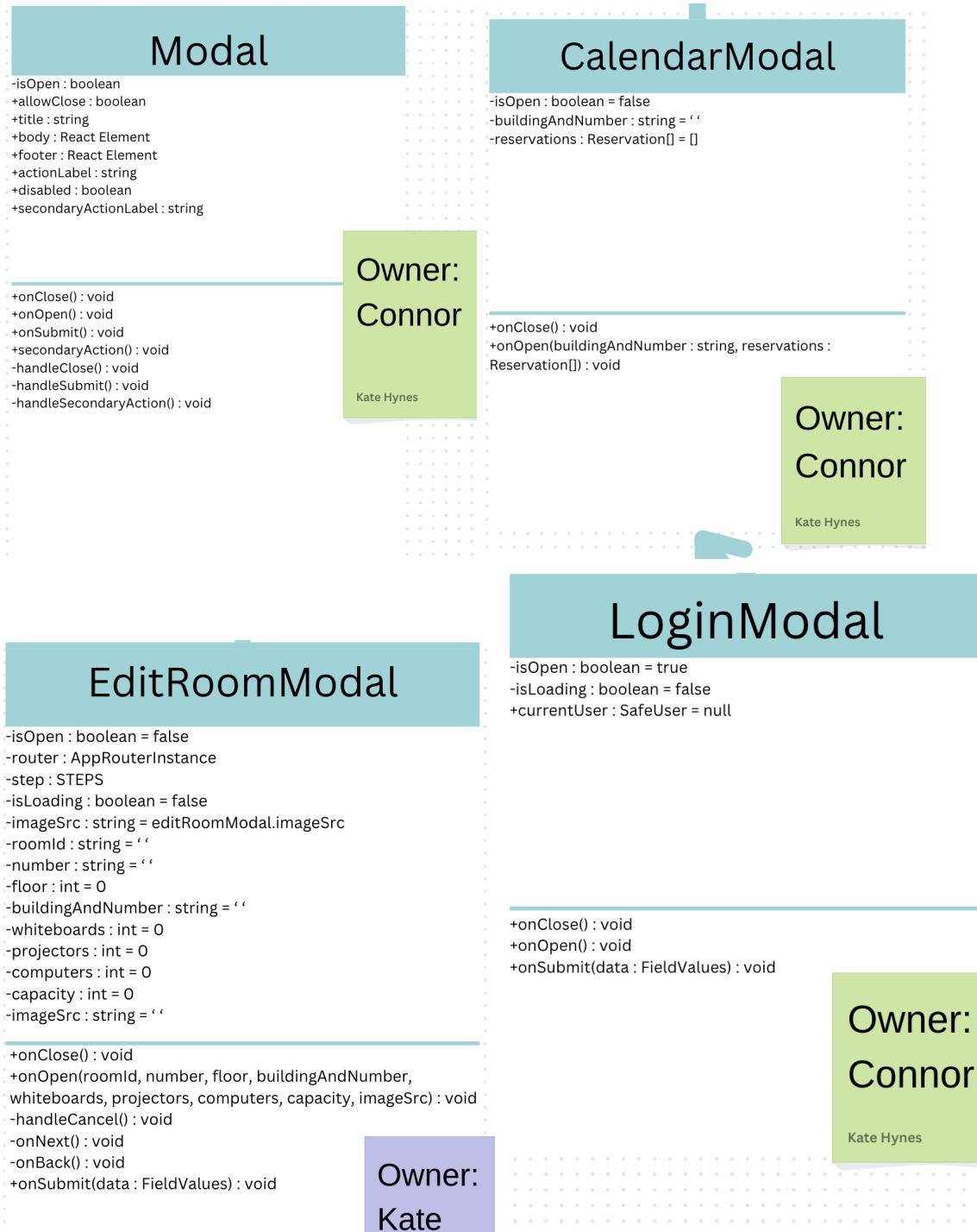
There are 3 images below that show the relationships between each class. The diagram is too large to fit on one image, so some elements in the diagram might be repeated across images. Below these images is a section with close-up images of each class in the diagram to show their relevant attributes and methods. Note that actions such as getCurrentUser() are functions that are exported separately, so they are given their own class diagram, but parentheses are included to denote that they are functions. Generalizations are indicated by open arrows, dependencies are indicated by dashed lines, and associations are indicated by solid lines

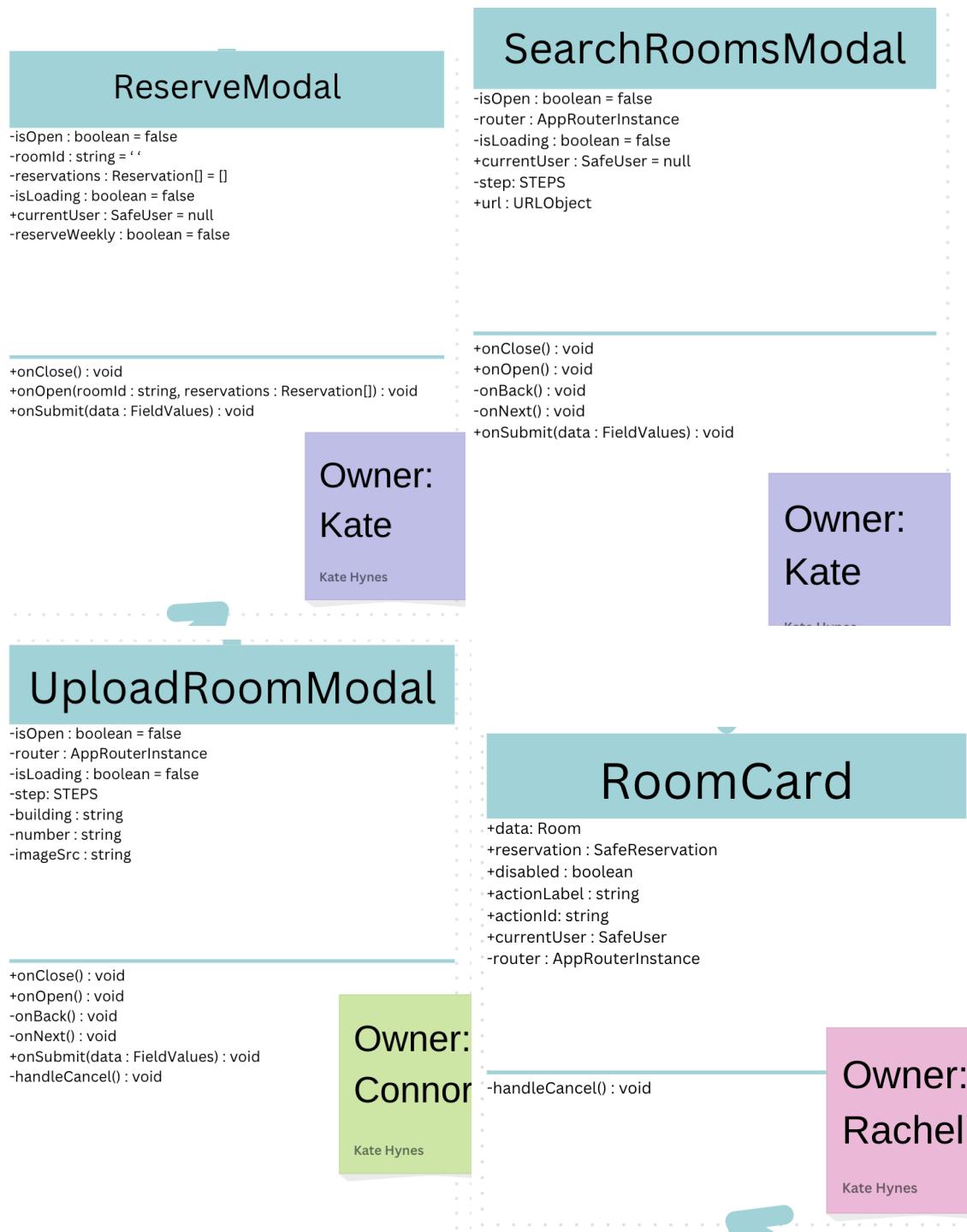


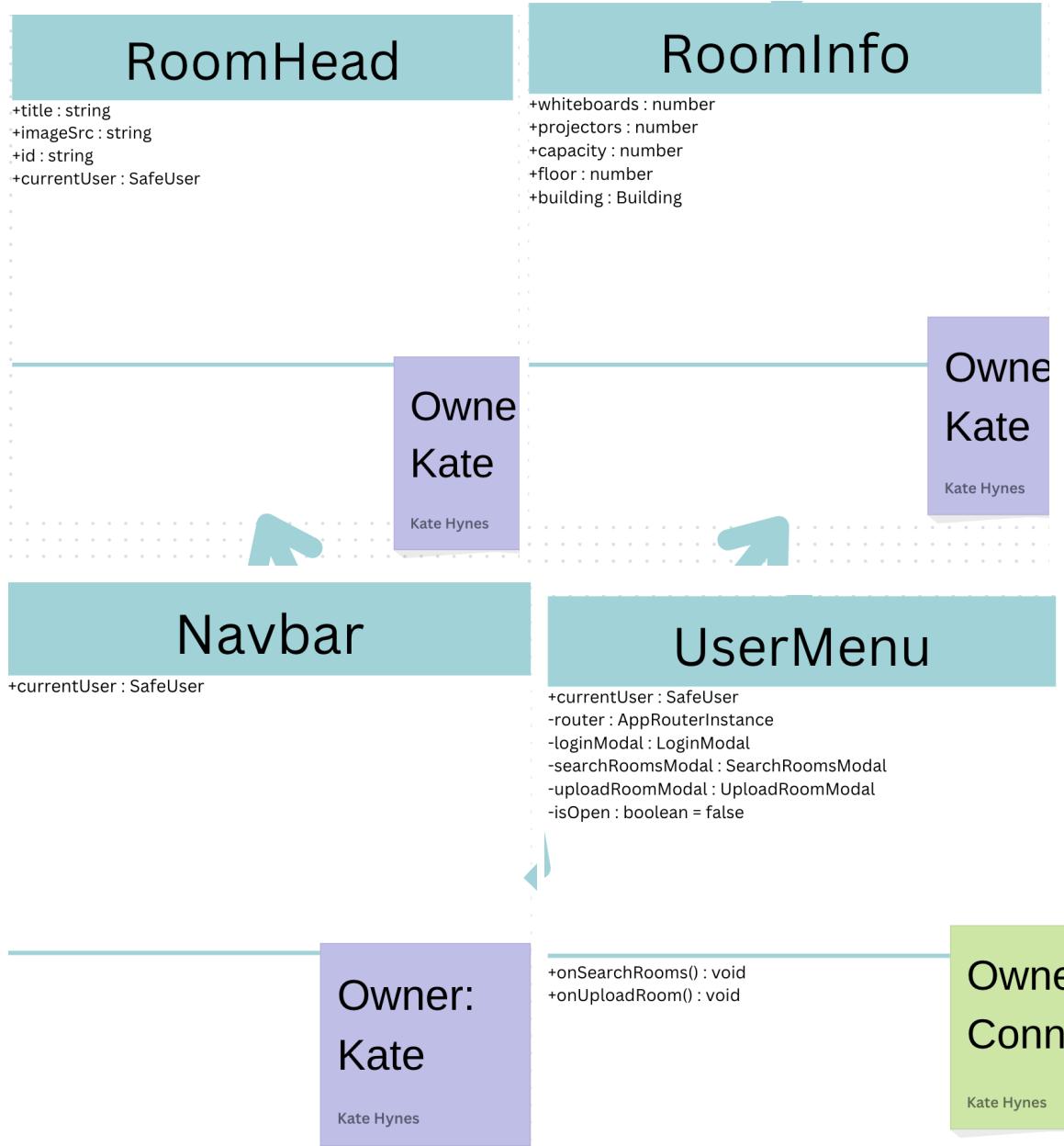


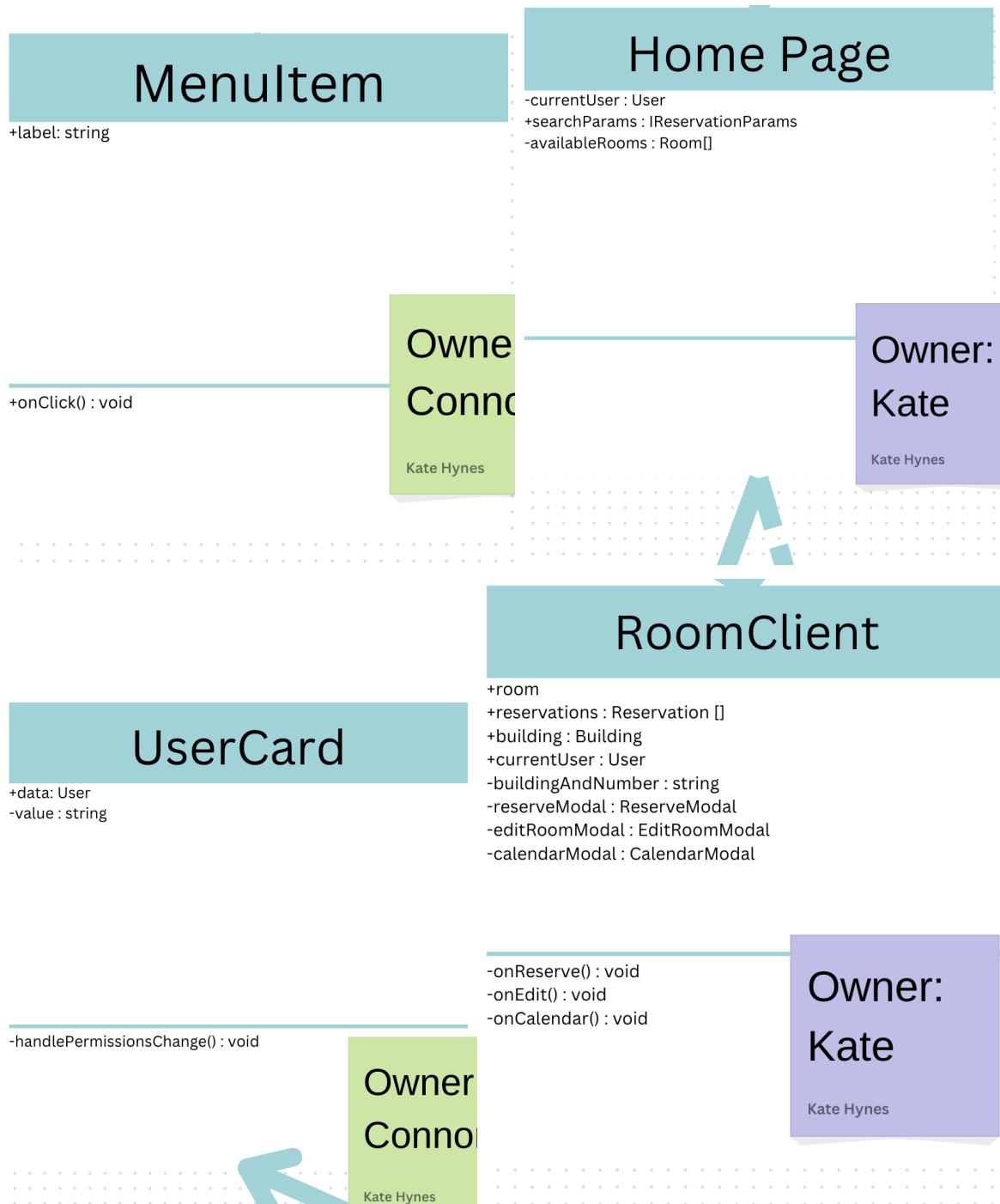


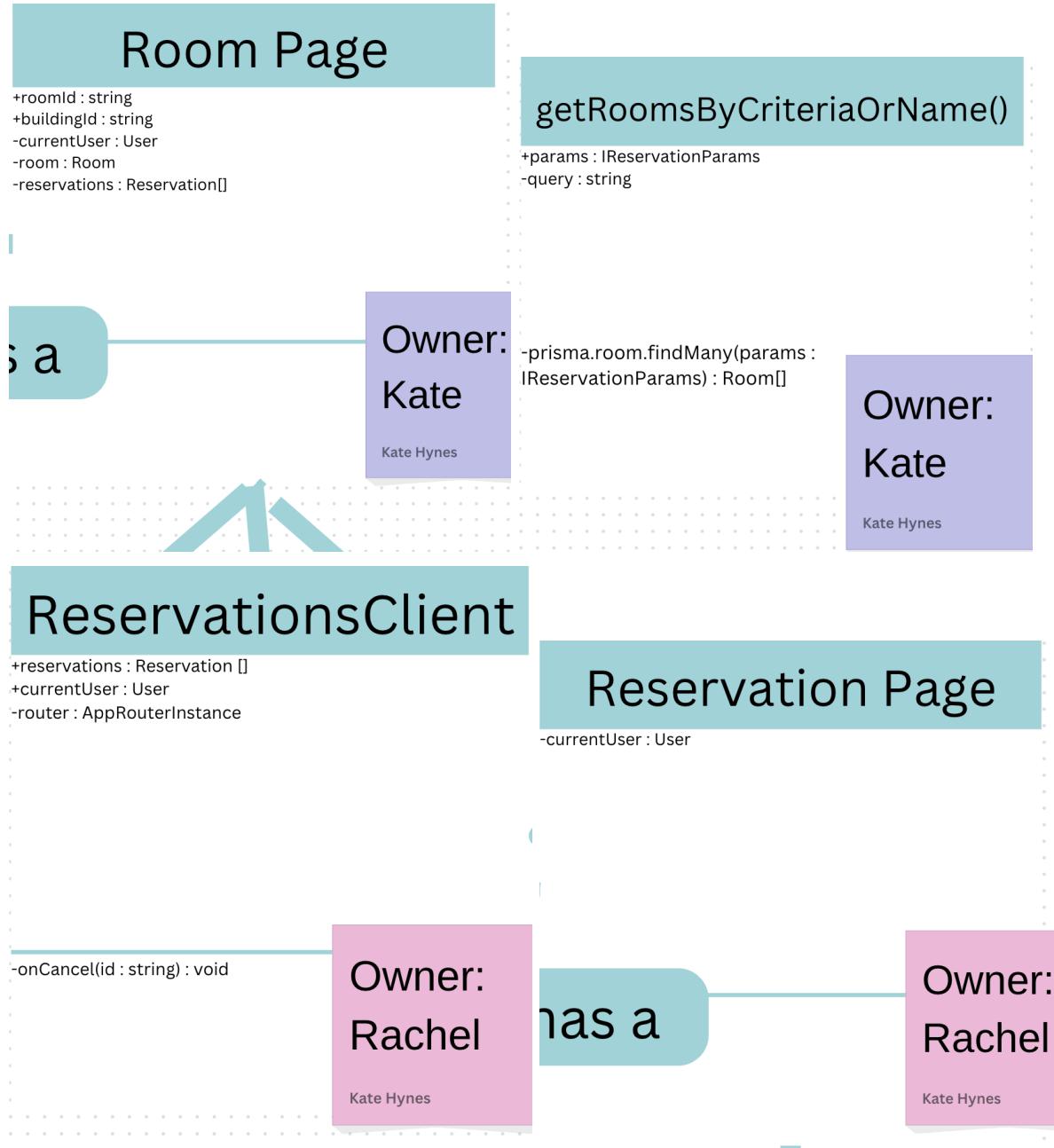
Individual views of each class in the diagram.











Favorites Page

```
+searchParams : IReservationParams
currentUser : User
favoriteRooms : Room[]
```

PermissionsClient

```
+currentUser : SafeUser
+users : User[]
```

Owner:
Yumna
Kate Hynes

Owner:
Connor
Kate Hynes

Permissions Page

```
+roomId : string
+buildingId : string
currentUser : User
users : User[]
```

getRoomById()

```
+roomId : string
```

Owner:
Connor
Kate Hynes

Owner:
Connor
Kate Hynes

```
-prisma.room.findUnique() : Room
```



7. Current Status

a. Summarize the current implementation status of your system.

We have successfully implemented all targeted features in the current system. Our system includes a functional login page that utilizes the Google OAuth2 API for user authentication. If a user is not already registered, the system automatically creates a new user entry in our database. Our interface features a navbar that allows users to return to the home page by clicking an icon and also greets them with a personalized welcome message. The navbar provides options to search for rooms based on criteria such as time availability, room number, and the presence of projectors, whiteboards, and computers. For users with admin permissions, there is an added functionality to upload room details through a modal accessible via the navbar. The navbar also includes a user menu for navigating to various pages on our site. The home page displays all rooms from the database in their own cards, with options to mark rooms as favorites. The favorites page, accessible through the navbar, showcases all favorited rooms for a given user. Additionally, there is an upcoming reservations page that lists all future reservations linked to the user, providing options to cancel reservations as needed. Clicking on any room redirects the user to a detailed page about that room, where they can make reservations and view the room's schedule. Users with admin permissions, as well as elevated students and professors, can edit room details and set up recurring reservations for a semester. Exclusive to admins, there is a permissions page where they can manage user access levels by searching through all registered users and adjusting their permissions to either expand or restrict their access to the system.

b. Screenshots

- i. *Add the screenshot(s) of the working system. Add the screenshots in the sequence of actions. This means that your screenshots will show you the flow of how the user interacts with the system. Think of it as how you would walk through the live demonstration during the presentation. Add the following details for each screenshot: 1. What are you trying to show in the screenshot? 2. Which part in the system description diagram does the part belong to? 3. What are the input and output of the system? 4. Why is this part of the system important?*
- ii. Login with Google API