Report #1

Group 9

14:332:452 Software Engineering

Gabriel Ajram

Omar Faheem

Gregory Giovannini

Nathaniel Glikman

Gaohaonan He

Yanbo Jiang

Hrishit Joshi

Oleksandar Kalynyak

Alex Malynovsky

Jake Seary

Github: https://github.com/greg300/team9

Table of Contents

Contributions Breakdown

All members contributed equally.

Customer Problem Statement

Problem Statement

Although the thermostat is a prized invention that has been in use for over a century, humidity is also an important facet of environmental health for our homes, plants, as well as ourselves. Humidity represents the concentration of water vapour in the air, and having too little or too much of it in a closed habitat can cause a plethora of problems. Too little humidity can cause electronic equipment to malfunction and health issues to arise, such as a dry throat or breathing problems. On the other hand, too much humidity creates a breeding ground for various bacteria and fungi to grow, causing moldy bathrooms, musty smells, potential allergic reactions, and overall damage to the residence. Additionally, plants require a great amount of attention when it comes to maintaining a healthy environment. Factors within the environment that can affect plant health include humidity, temperature, and circulation. Controlling these factors can prevent fungal diseases from surfacing, such as mildew and botrytis that can cause significant plant damage. Humidity is a primary culprit for a plant's health and ultimate lifespan, and being able to detect and control its parameters should be an option available to anybody, whether it be a plant hobbyist or a greenhouse enthusiast on a larger scale.

Today, the typical home-owner does not own a device to measure the RH (relative humidity) of their surroundings. These devices are called hygrometers and are very cheap to obtain and effortless to use, capable of saving money for environmental damage and health problems in the long term. Since humidity is an issue that scales across all environments, whether it be residences or businesses such as a greenhouse, we would like to implement an application that monitors the environment and interacts with the user in real-time in regards to their place of choosing.

Typical hobbyists who grow a few plants within their own home will likely not possess the space or equipment necessary for a very meticulous care routine. At most, they may own a humidifier and dehumidifier, with no special means of controlling either aside from turning them on or off at their own discretion. The plants in their house could be spread across multiple rooms, making it difficult to remember to water them when needed. On top of that, the plants may be sharing space with small children in the house, who can be sensitive to moisture (or lack thereof) in ways that must be prioritized. Moreover, when going away on a trip or vacation, the house is usually left to regulate itself in terms of humidity, which would ultimately be detrimental for plants. Additionally, the average person is simply unaware of the ideal humidity parameters for the growth of any particular plant. The research that goes into finding such answers is not trivial, and depending on the specific needs of that household, the answers they do find may not even be what is truly best for them.

On the other hand, greenhouse owners who maintain entire gardens' worth of plants in a controlled environment likely have a much more dedicated setup. Plants tend to be arranged logically and without the same degree of interference from other external factors that hobbyists need to worry about. However, there may be varieties of different plants that require an assortment of unique hydration, and managing the humidity for them all, especially when away from the greenhouse, can be tricky to automate. Even if someone is technically capable of managing this themselves, it is doubtlessly a headache to do so, and anything that can expedite this process is surely a welcome addition to the gardening world.

Our system will comprise of two modes of operation: Residence and Greenhouse Mode. Both modes will give the user the ability to create their environment and customize any external parties that can affect the room's status. The Residence Mode will have simple features only a typical home-owner would need, such as humidity levels, recommended courses of action, and proper calibration of their hygrometer. Greenhouse Mode will accommodate more advanced tweaking and configuration, such as creating a personalized plan based on various additional parameters typically included in a greenhouse. Today, the remedies in place to regulate these aspects are vastly inefficient and difficult to implement and uphold in a controlled greenhouse. Additionally, some of these methods such as a vent and heating system can become quite costly. These practices include:

- Misting
- High Density Grouping
- Smart Watering
- Bottom Heat

- Anti-drip Plastic
- Ventilation and Heating

Simply connecting and interfacing with an AC unit or a hygrometer does not approach the level of customizability and automation our system proposes. Instead, we seek to provide a flexible tool with which users can monitor and control the relative humidity of the environment for plants and family. A user will have the ability to save data associated with their home or greenhouse - existence of humidity-sensitive family members for a Residence, number and type of plants, arrangement of plants, necessary hydration practices for certain plants in a Greenhouse Mode.

Our system will be able to intake these options and monitor as well as recommend a plan of action in order to satisfy your plants' needs. The system will have access to a database that will hold data specific to you. That information will include things such as the types of plants you own, the relative quantities of these types, the arrangement of the plants, and a user's preferences for specific rooms he or she has the system set up in.

Moreover, a significant part of plant care (and home care) is communicating and collaborating with other plant/home-owners. Thus, it makes sense to include a discussion forum with which users can pose questions, share solutions and remedies, and exchange information with one another. The forum itself would be community-driven and didactically-oriented, with the intention of having more experienced plant-owners helping newer ones work through any problems. This should be integrated with the application and be accessible to all users, regardless of their Mode (Residence or Greenhouse).

We believe that our system will appeal to people looking for an easy way to track their plants' needs, and even supply them with the tools to automate their home life. One benefit of having the system cater to the user's preferences is the ability to not focus on plants. While there is a strong focus on helping automate plant care, the system can be used to simply track the humidity levels of your home, and can adjust on the fly whenever you need it to. Or, as discussed earlier, the presence of children or humidity-sensitive people in a home can be more effectively treated by calibrating the system to their needs. This way, the system has a much wider appeal to people in general, thus increasing the user base. Additionally, it can pique interest in plant care;

an indoor gardening novice who is just getting the hang of things will find our system of the utmost importance, as it will ease the difficulty of navigating a whole new world of variables. The diverse array of applications that this has will ultimately provide a large ecosystem of users tracking humidity information in their homes.

Glossary of Terms

RH: relative humidity refers to the concentration of water vapor in the air. As opposed to other techniques of measuring humidity, this method allows us to simplify our calculations in making clear and concise recommendations to the end-user.

Residence Mode: Offers a collection of more user-friendly features to your average home-owner, omitting more advanced recommendations and room parameters.

Greenhouse Mode: Offers everything Residence Mode has to offer in addition to more advanced features that would be familiar to any greenhouse manager.

Room: The user will be able to recreate their own collection of rooms that they want to monitor.

External Parties: These include factors that can help/affect the humidity of the current room. There are passive elements such as windows and doors as well as active elements such as humidifiers. These factors expand into more advanced features in the Greenhouse Mode.

Dashboard: Refers to the main area of the system where the user can view all of their personalized rooms' latest sensory readings.

Room View: Refers to the page that opens that retrieves that specific rooms' readings and elements.

Away Mode: App not currently open.

Active Mode: App is currently open.

Personalized Plan: Refers to the plan available to view in Room View that updates on every new element addition and is widely dependent on plant type if in Greenhouse Mode.

System Requirements

Enumerated Functional Requirements

Priority Scale: 1 – 5, with 5 being highest priority.

Identifier	Priority	Requirement	Acceptance Test Cases
REQ-1	5	The user should be able to register an account with our database with an email, a username, and a password.	Test with users signing up with their email and creating a password.
REQ-2	5	The user should be able to log into an account with their registered username and password.	Test with a user, once signed up, successfully logging into the system.
REQ-3	2	The user should be able to request a password reset on the login page.	Test with a user successfully changing a password after clicking the “Forgot Password” button on the login page.
REQ-4	4	The user should be able to seamlessly switch between Residence Mode and Greenhouse Mode.	Test with toggle and whether or not user parameter history is carried over.
REQ-5	4	The user should be able to create additional virtual ‘rooms’ in the app.	Test with creating more rooms, and making sure each users’ rooms store independent customizations by the user.
REQ-6	3	The user should be able to customize each room with available parameters depending on the current active mode.	Test with adding parameters to a room in Residence Mode and in Greenhouse Mode. Verify that adding parameters updates recommended

			humidity settings.
REQ-7	3	The user should be able to remove a Room from their account.	Test with adding a Room and subsequently deleting it, making sure the data associated with it has been properly erased from the database.
REQ-8	3	The user should be able to post to an integrated discussion forum regarding situations with their plants or home’s humidity.	Test with a user making a post and another user responding to the post.
REQ-9	1	The user should be able to modify or remove a post from the discussion forum.	Test modifying a post. Test deleting a post.

Enumerated Nonfunctional Requirements

Priority Scale: 1 – 5, with 5 being highest priority.

Identifier	Priority	Requirement	Acceptance Test Cases
REQ-10	4	When on Away Mode, the system should do a regulatory readings update in the rooms every 10 minutes.	Test the system while in Away Mode and observe that the system does a check and records data every 10 minutes.
REQ-11	4	When in Active Mode, the system should update the sensory readings every 10 seconds.	Test the system while in the system and ensure that readings are updated every 10 seconds.
REQ-12	3	Make the login page less vulnerable to common Cyber-attacks like Buffer overflow.	Test the defense mechanism of the login page with a low intensity buffer overflow attack.
REQ-13	2	The system should presume its default day/night cycle when the user logs out or enters 'Away Mode'.	Test the system while on Away-mode and observe that the system recognises the day/night cycle.
REQ-14	2	The system should remember past settings in order for it to be faster.	Test the memorability of the system by successfully accessing past settings.
REQ-15	5	The Personalized Plan in the Room View should update on every new element addition but not on sensory readings.	Test with new element addition and sensory reading updates.
REQ-16	3	The login page should have two-factor authentication.	Test logging in and successfully passing the 2-factor authentication test.
REQ-17	5	The system should be able to detect and interpret extreme readings above or below a certain threshold to alert the user of potentially unfit/unsafe conditions	Test by feeding the sensors extreme input and seeing how the system reacts to the user.

User Interface Requirements

Priority Scale: 1 – 5, with 5 being highest priority.

Identifier	Priority	Requirement	Acceptance Test Cases
REQ-18	5	The User Interface should provide a button to switch between Residence and Greenhouse mode	Test with user being able to locate the button to switch between modes
REQ-19	5	The User Interface should provide a button that would get the menu open	Test with user being able to locate the button to get the menu open
REQ-20	5	The User Interface should provide a Button to add or remove the greenhouse data or room data	Test with user being able to locate the button to get the menu open
REQ-21	3	The Login Page should be structured such that it is usable	Have a user test it and record and observe how it meets the 5 requirements of usability.
REQ-22	4	Each room should have the ability to be clicked on to view its independent readings and elements.	Test with user being able to click each separate room and retrieve separate readings from each physical place.
REQ-23	5	Each Room View should have a personalized plan that caters to the rooms' elements, readings and potential plant type.	Test with user being able to obtain personalized course of action that depends on their readings and element input.
REQ-24	3	The menu should be populated with a collection of pages such as Account Info, a 'How it Works' page, a FAQ page, and an option to Log Out of the current account.	Test that the menu is populated with said pages, and that the pages load, or that the functions work.
REQ-25	3	Each Room on the Dashboard should display immediate readings for QoL purposes, so that the user does not have to click on each room to view readings.	Test that the readings on each room in the dashboard display and update properly.

REQ-26	4	Each Room should have the ability to highlight red or display a red exclamation mark inside it to alert the user of potentially unfit/unsafe conditions	Test with feeding the sensors extreme input to make sure the system is able to catch and alert the user.

Functional Requirements Specification

Stakeholders

Plant Hobbyists, Homeowners, Greenhouse Farming Companies

Actors and Goals

User - The user can access all front end facets of the web application, including logging in, setting up their humidifier settings, and will be the one setting up the humidifier and dehumidifier with the arduino to connect to the web application

Web App - The web app will be able to connect to both the arduino setup and the database with credentials and information of the user. The user will be able to interact with the humidifier setup through the web app

Database - Houses login credentials and user preferences to called upon when needed

Hygrometer - Takes humidity readings for the arduino

Thermostat - Takes temperature readings for the arduino

Arduino - Interacts with both the hygrometer and the thermostat. Receives commands from the web app for changing humidity levels. Reads information from the hygrometer and thermostat that's also connected and sends that information to the web app for displaying to the user

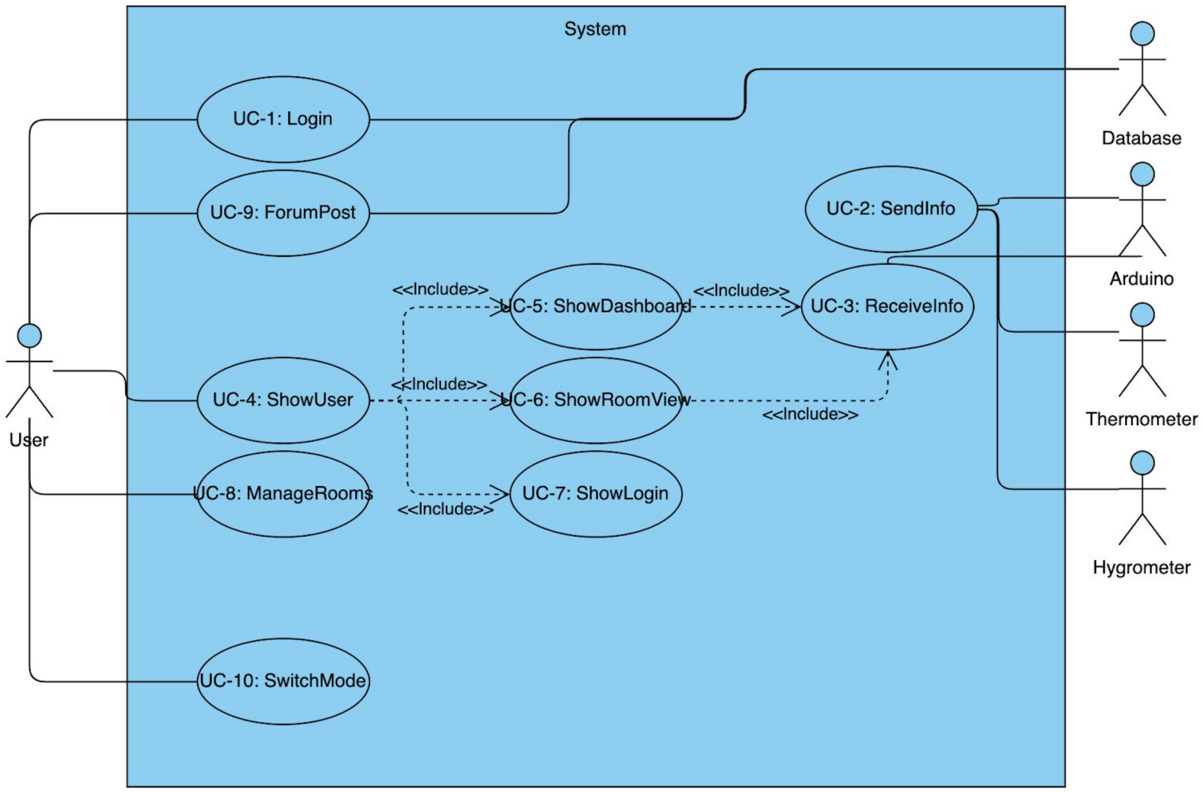
Use Cases

Casual Description

Use Case Name	**Actor**	**Actor's Goal**	**System Requirements**
UC-1: Login	User, Web app, Database	To allow access to the app to the specific user.	REQ-1, REQ-2, REQ-3, REQ-12, REQ-16, REQ-21
UC-2: SendInfo	Web app, Arduino, Hygrometer, Thermostat	To send information to the web app when called upon.	REQ-10, REQ-11
UC-3: ReceiveInfo	Web app, Arduino, User	To receive information from the arduino and display to the user.	REQ-10, REQ-11, REQ-17, REQ-22, REQ-25, REQ-26
UC-4: ShowUser (includes ShowDashboard, ShowRoomView, ShowLogin)	User, Web app	To show user information from different actors and display them to the web app based on where the user is on the web app.	REQ-1, REQ-2, REQ-3, REQ-21, REQ-22, REQ-23, REQ-25
UC-5: ShowDashboard (includes ReceiveInfo)	User, Web app	To build and display the Dashboard for viewing all user's Rooms' latest sensory data.	REQ-25
UC-6: ShowRoomView (includes ReceiveInfo)	User, Web app	To build and display a view of a given Room's elements, stats, recommended course of action, etc.	REQ-22, REQ-23
UC-7: ShowLogin	User, Web app	To display the login page to a user trying to access the app.	REQ-1, REQ-2, REQ-3, REQ-21

UC-8: ManageRooms	User, Web app	To add a new Room for a user, allowing them to specify parameters and external parties in either Greenhouse or Residence Mode; to remove or modify an existing Room.	REQ-5, REQ-6, REQ-7
UC-9: ForumPost	User, Web app, Database	To add, modify, or delete a post to the web app's integrated blog.	REQ-8, REQ-9
UC-10: SwitchMode	User, Web app	To switch the current mode between Greenhouse Mode and Residence Mode.	REQ-4

Use Case Diagram



UC-4 is the big use case because of how important the UI is with regards to showing data, so we want to break it up into smaller use cases beyond the general "ShowUser".

Traceability Matrix

Req	**PW**	**UC-1**	**UC-2**	**UC-3**	**UC-4**	**UC-5**	**UC-6**	**UC-7**	**UC-8**	**UC-9**	**UC-10**
1	5	X			X			X			
2	5	X			X			X			
3	2	X			X			X			
4	4										X
5	4								X		
6	3								X		
7	3								X		
8	3									X	
9	1									X	
10	4		X	X							
11	4		X	X							
12	3	X									
13	2										
14	2										
15	5										
16	3	X									
17	5			X							
18	5										
19	5										
20	5										

21	3	X			X			X			
22	4			X	X		X				
23	5				X		X				
24	3										
25	3			X	X	X					
26	4			X							
Max PW		5	4	5	5	3	5	5	4	3	4
Total PW		21	8	24	27	3	9	15	10	4	4

Fully-Dressed Description

Use Case UC-2: SendInfo	
Related Req.:	REQ-10, REQ-11
Init. Actors:	Arduino
Actor's Goal:	To send information to the web app when called upon
Participating Actors:	Hygrometer, Thermostat, Web App
Preconditions:	System is already active and running in the background
Postconditions:	To receive information from the arduino and display to the user
Flow of Events for Main Success Scenario	1. The Hygrometer and Thermostat feed the data into the webapp through Arduino. 2. The webapp updates information according to the mode the system is in. 3. If Mode is active, the data is updated every 10 seconds. 4. If Mode is Away, the data is updated every 10 minutes.

Use Case UC-3: ReceiveInfo	
Related Req.:	REQ-10, REQ-11, REQ-17, REQ-22, REQ-25, REQ-26
Init. Actors:	Web app
Actor's Goal:	To receive information from the arduino and display to the user
Participating Actors:	Arduino, User
Preconditions:	Arduino is connected and has information to send
Postconditions:	Web app receives arduino information
Flow of Events for Main Success Scenario	1. The web app requests information from the arduino. 2. The arduino sends the web app its latest information via SendInfo.

Use Case UC-4: ShowUser	
Related Req.:	REQ-1, REQ-2, REQ-3, REQ-21, REQ-22, REQ-23, REQ-25
Init. Actors:	User
Actor's Goal:	To show user information from different actors and display them to the web app based on where the user is on the web app.
Participating Actors:	Web app
Preconditions:	User has navigated to a new page on the web app.
Postconditions:	The page requested by the user will be displayed.
Flow of Events for Main Success Scenario	5. User requests a new page by navigating to it. 6. If page is Login Page, the user is shown the Login Page. 7. If page is Dashboard, the user is shown the Dashboard. 8. If clicking on a particular Room, the user is shown the Room View for that Room.

Use Case UC-8: ManageRooms	
Related Req.:	REQ-5, REQ-6, REQ-7
Init. Actors:	User
Actor's Goal:	To add a new Room for a user, allowing them to specify parameters and external parties in either Greenhouse or Residence Mode; to remove or modify an existing Room.
Participating Actors:	Web app
Preconditions:	User is in Dashboard if adding a new Room; if removing or modifying a Room, there is a Room being targeted in Room View.
Postconditions:	The requested action on a Room will be completed.
Flow of Events for Main Success Scenario	Adding a Room: 1. User clicks on button to create a new Room. 2. User enters Room name and specifies Room parameters,

detail of which is determined by whether they are in Greenhouse or Residence Mode.

3. Room is added to user’s account in the database.
4. User is taken to the Room View page for the new Room.

Removing a Room:

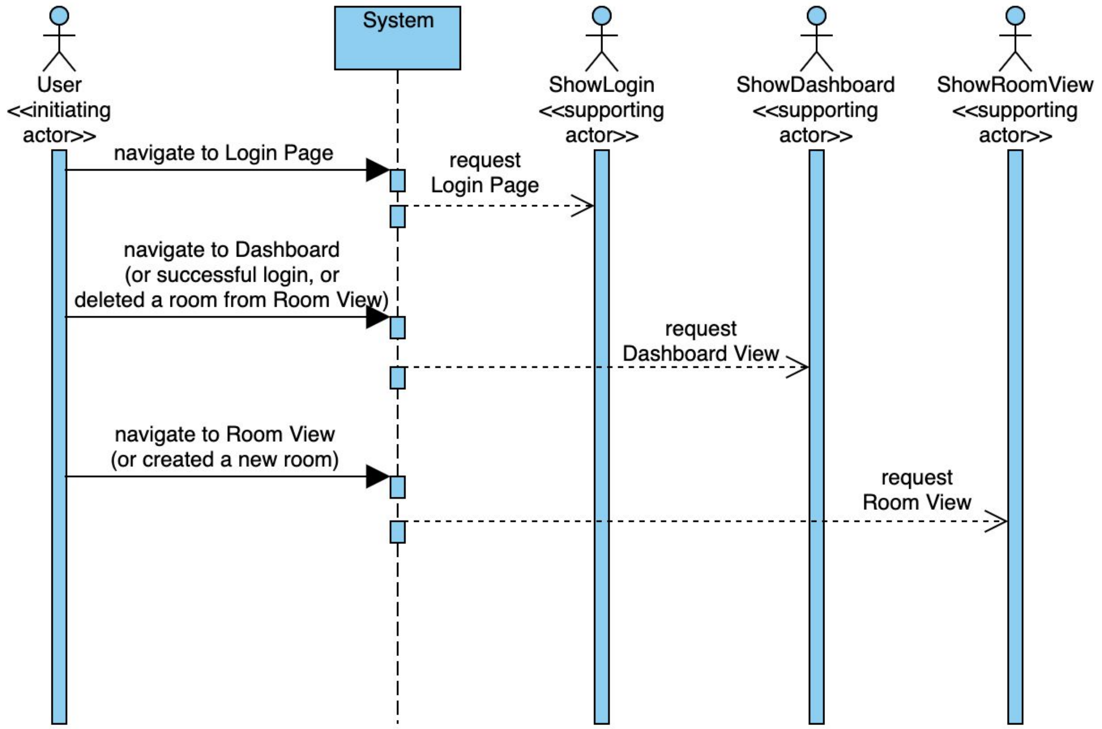
1. User clicks on button to delete the given Room.
2. An alert pops up to request confirmation of deleting the Room.
3. If user declines, the Room is not deleted and the popup disappears.
4. If user accepts, the Room is removed from the user’s account in the database, and the user is returned to the Dashboard.

Modifying a Room:

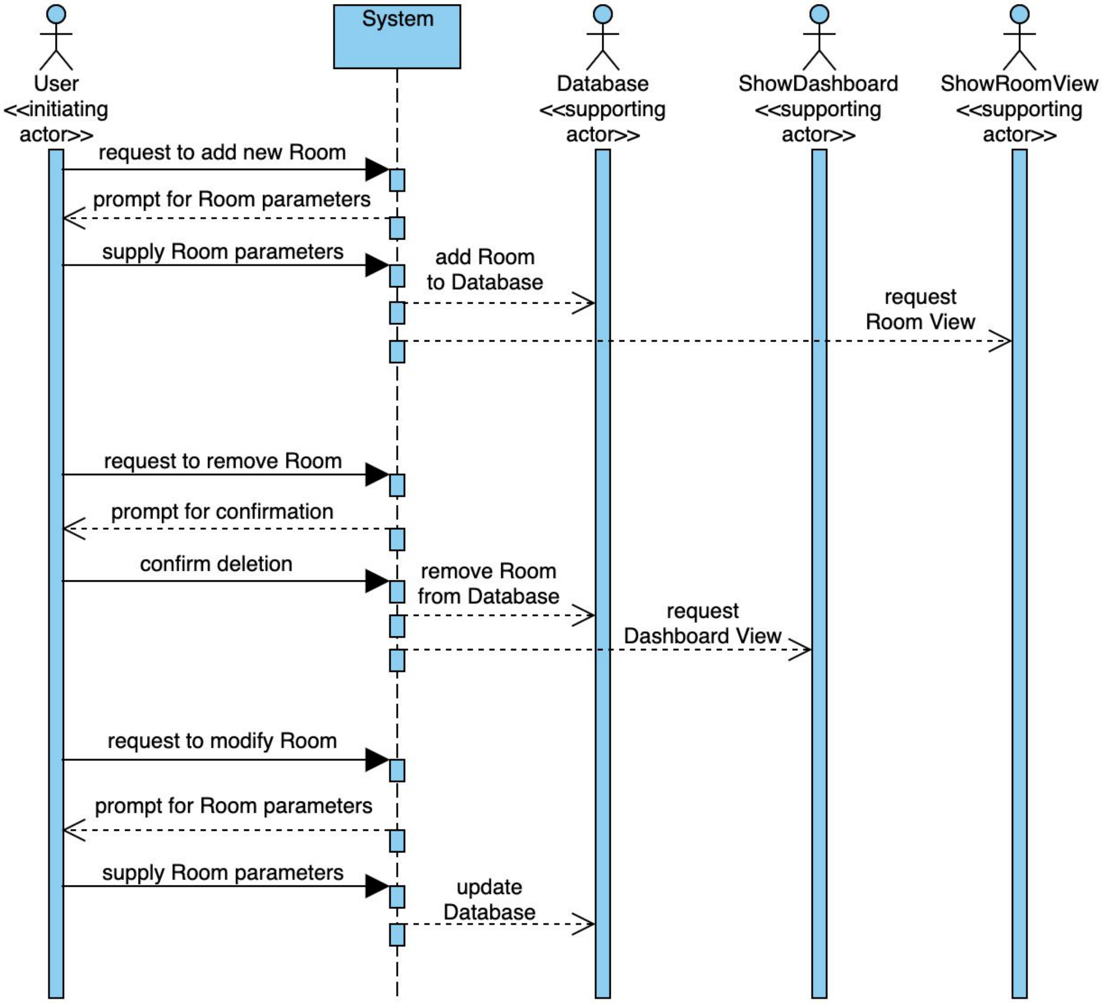
1. User clicks on button to modify a given Room.
2. User is shown their Room’s parameters and may add or remove as desired.
3. User confirms or rejects their own changes.
4. Database updates with changes.
5. Room View is updated with new parameters.

System Sequence Diagrams

Use Case UC-4: ShowUser



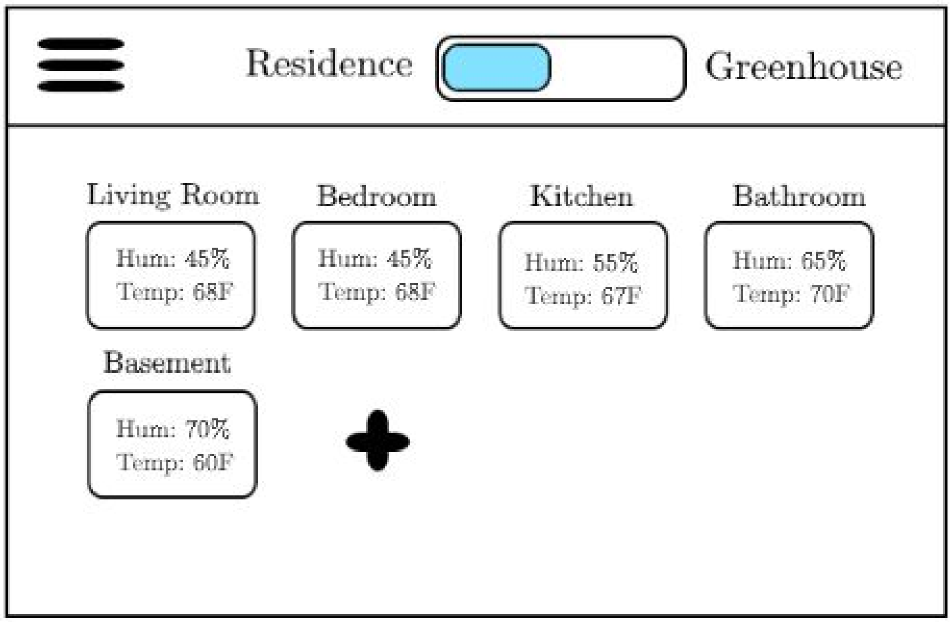
Use Case UC-8: ManageRooms



User Interface Specification

Preliminary Design

Dashboard View - Residence Mode



Living Room
Hum: 45%
Temp: 68F

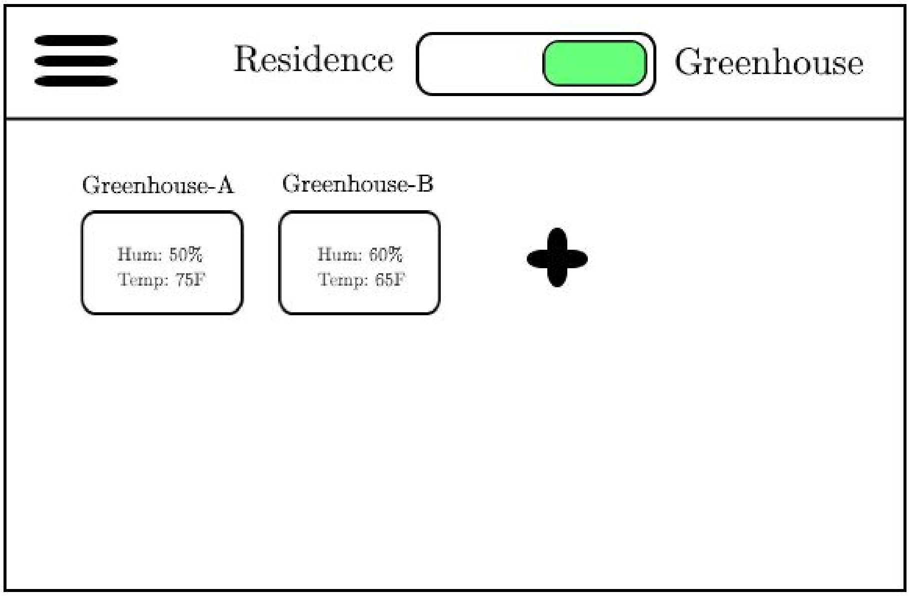
Bedroom
Hum: 45%
Temp: 68F

Kitchen
Hum: 55%
Temp: 67F

Bathroom
Hum: 65%
Temp: 70F

Basement
Hum: 70%
Temp: 60F

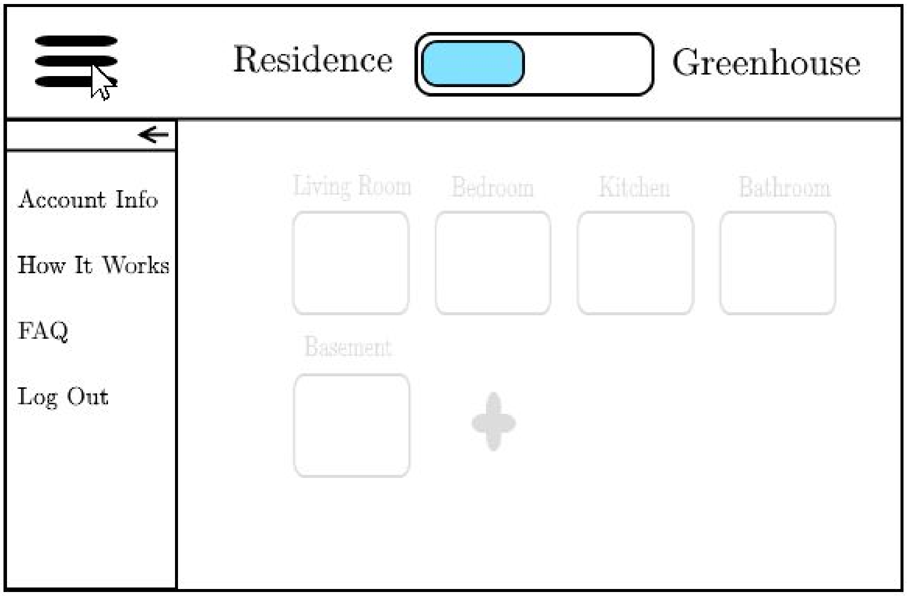
Dashboard View - Residence Mode



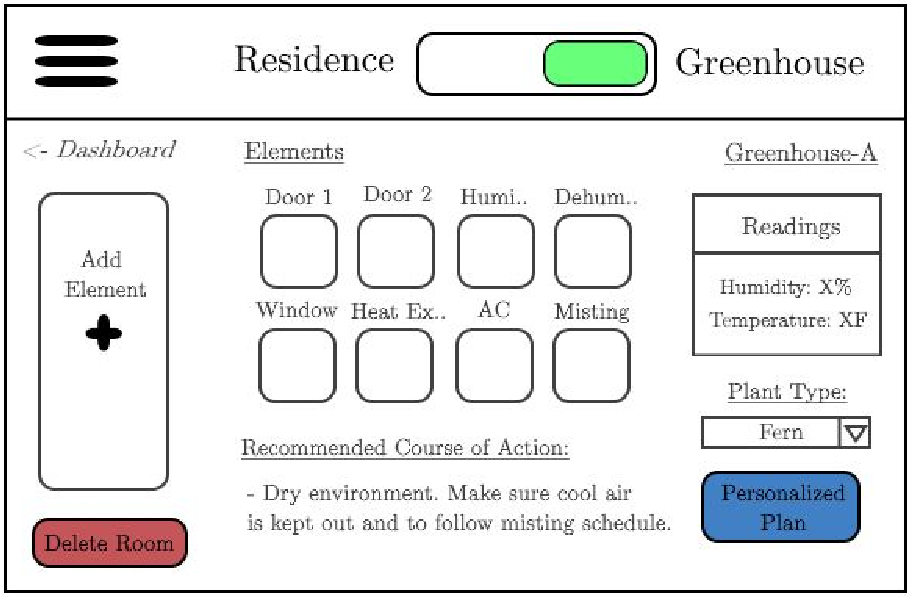
Greenhouse-A
Hum: 50%
Temp: 75F

Greenhouse-B
Hum: 60%
Temp: 65F

Navigation Menu



Room View



Humidity: X%
Temperature: XF

Recommended Course of Action:

- Dry environment. Make sure cool air is kept out and to follow misting schedule.

The Navigation Menu will be accessible from any page. When clicked, it will expand to allow the user to access their account info, a help page, an integrated discussion forum (FAQ), and to return to the login page (Log Out).

The “home page” is the Dashboard View. The user is redirected here upon login. From here, all current Rooms can be viewed as small icons and top-level stats for each room are displayed on those icons.

Clicking on a room in the Dashboard will take the user to a Room View page, where all elements and more detailed readings can be viewed. The option to add more elements or configure existing elements, as well as delete the room, are all available with one or two clicks. Returning to the Dashboard takes a single click of a return button.

User Effort Estimation

Adding a Room named “Test” with no starting parameters/elements:
Assume the user is already logged in and on the Dashboard View.

1. Navigation: total 2 mouse clicks, as follows
    a. Click on the “+”
    b. (Complete data entry as below)
    c. Click “Ok” to finish.
2. Data Entry: total1 mouse click and 4 keystrokes
    a. Click inside the “Name” box
    b. Press the 4 keys in the word “Test”

Removing a Room named “Test”:
Assume the user is already logged in and on the Dashboard View, and there exists a Room named “Test”.

1. Navigation: total 3 mouse clicks, as follows
    a. Click on the icon for the Room “Test”
    b. Click on the “Delete Room” button
    c. Click “Confirm”

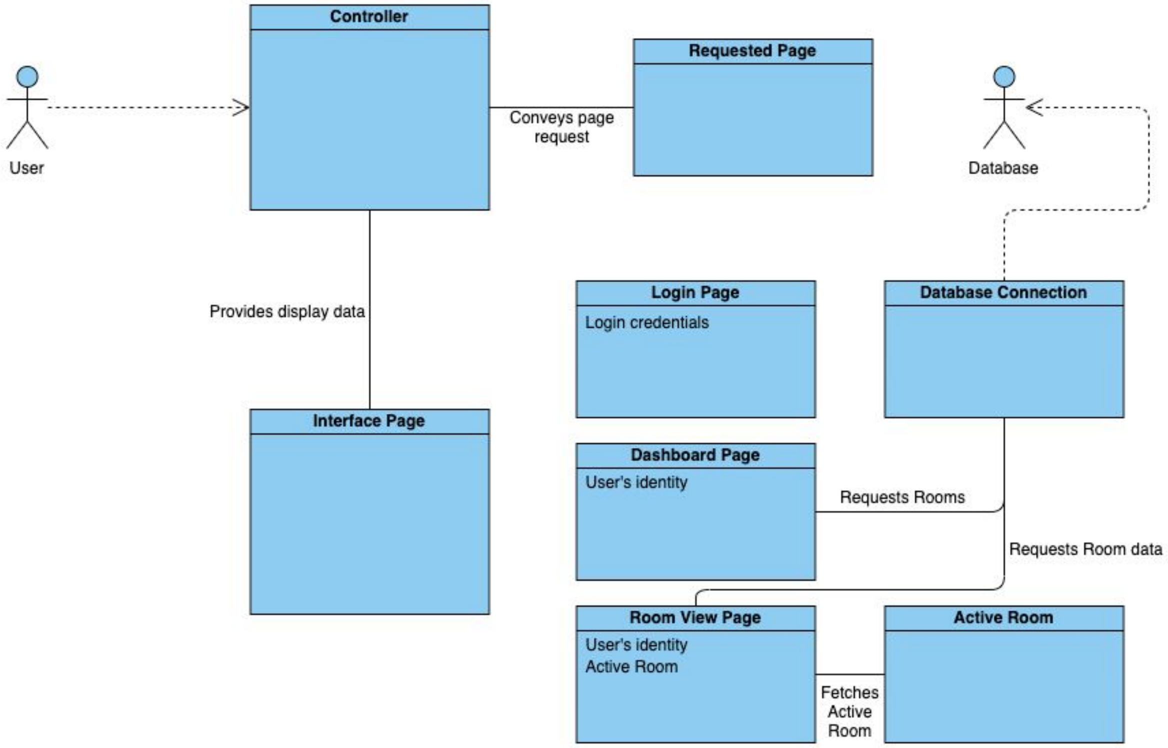
Add an element called “Door” to a Room named “Test”:
Assume the user is already logged in and on the Dashboard View, and there exists a Room named “Test”.

1. Navigation: total 3 mouse clicks, as follows
    a. Click on the icon for the Room “Test”
    b. Click on the “Add Element” button
    c. (Complete data entry as below)
    d. Click “Confirm”
2. Data Entry: total1 mouse click and 4 keystrokes
    a. Click inside the “Name” box
    b. Press the 4 keys in the word “Door”

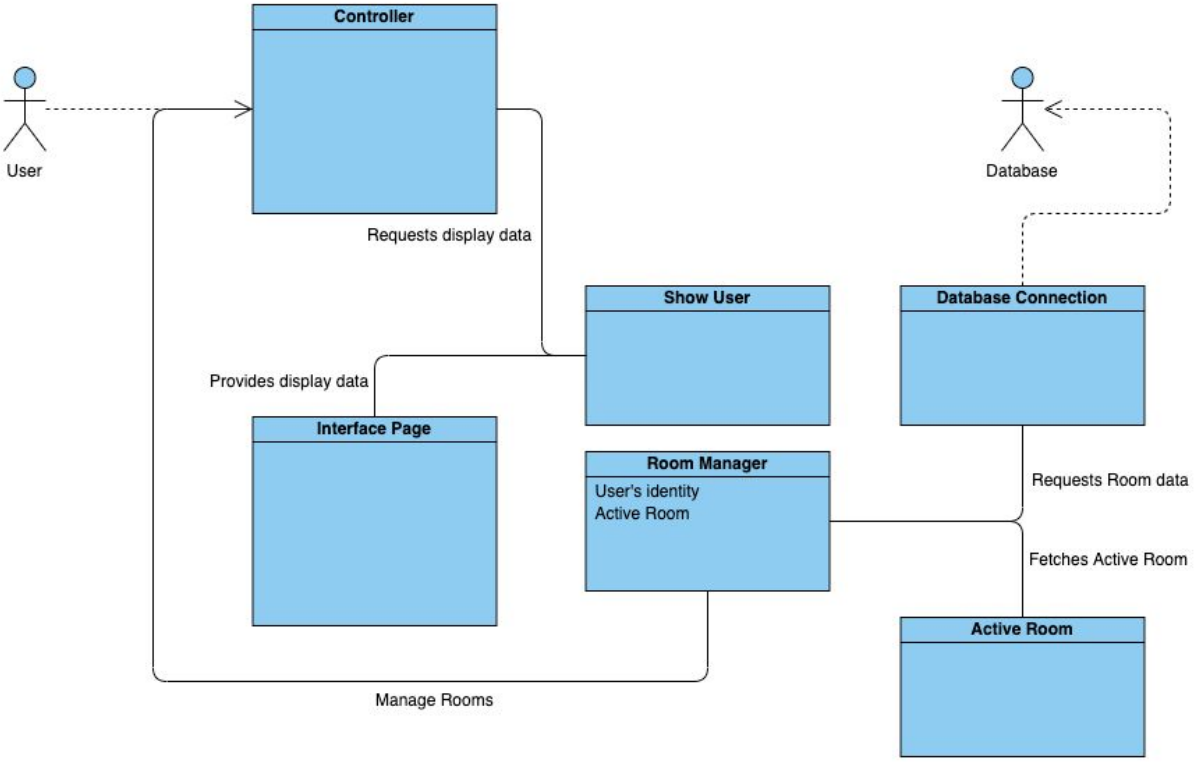
Domain Analysis

Domain Model

Use Case UC-4: ShowUser



Use Case UC-8: ManageRooms



Concept Definitions

Use Case UC-4: ShowUser

Responsibility Description	**Type**	**Concept Name**
Rs1. Coordinate actions of all concepts associated with this use case and delegate the work to other concepts.	D	Controller
Rs2. HTML document that shows the actor the current context, what actions can be done, and outcomes of the previous actions.	K	Interface Page
Rs3. Render the Login Page to display to the User.	D	Login Page
Rs4. Render the Dashboard Page to display to the User.	D	Dashboard Page
Rs5. Render the Room View Page to display to the User, based on the Active Room.	D	Room View Page
Rs6. Prepare a database query to retrieve the proper data associated with the User's given Room if requesting the Room View Page, or to retrieve the User's Rooms if requesting the Dashboard Page.	D	Database Connection
Rs7. Room that is currently selected by the User, if any.	K	Active Room
Rs8. The requested page to show the User, based on previous actions (e.g. entering the web page for the first time –> Login Page, adding a new Room –> Room View Page).	K	Requested Page

Use Case UC-8: ManageRooms

Responsibility Description	**Type**	**Concept Name**
Rs1. Coordinate actions of all concepts associated with this use case and delegate the work to other concepts.	D	Controller
Rs2. HTML document that shows the actor the current context, what actions can be done, and outcomes of the previous actions.	K	Interface Page
Rs3. Handle adding, removing, and modifying new Rooms.	D	Room Manager
Rs4. Prepare a database query to add, remove, or modify a Room.	D	Database Connection
Rs5. Update the Interface Page with the Room View Page when adding or modifying a Room, or the Dashboard Page when removing a Room.	D	Show User
Rs6. Room that is currently selected by the User, if any.	K	Active Room

Association Definitions

Use Case UC-4: ShowUser

Concept Pair	**Association Description**	**Association Name**
Controller – Interface Page	Controller populates the Interface Page with the proper page to display.	Provides display data
Controller – Requested Page	Controller uses the given Requested Page to determine whether to build a Login Page, a Dashboard Page, or a Room View Page.	Conveys page request
Room View Page – Active Room	Room View Page builder utilizes the targeted Active Room to determine which existing Room should be queried for in the Database.	Fetches Active Room
Room View Page – Database Connection	Room View Page requests Database Connection to fetch data for the Active Room.	Requests Room data
Dashboard Page – Database Connection	Dashboard Page requests Database Connection to fetch all associated Rooms.	Requests Rooms

Use Case UC-8: ManageRooms

Concept Pair	**Association Description**	**Association Name**
Controller – Show User	Controller requests the proper page to be displayed.	Requests display data
Show User – Interface Page	Show User populates the Interface Page with the proper page to display.	Provides display data
Controller – Room Manager	Controller requests to add, remove, or modify a given room.	Manage Rooms
Room Manager – Active Room	Room Manager utilizes the targeted Active Room to remove or modify it.	Fetches Active Room
Room Manager – Database Connection	Room Manager queries the Database to remove or modify the Active Room, or add a new one.	Requests Room data

Attribute Definitions

Use Case UC-4: ShowUser

Concept	**Attributes**	**Attribute Description**
Login Page	Login credentials	Credentials entered by the actor upon trying to log in.
Dashboard Page	User's identity	Used to determine the actor's credentials, which in turn specify what data this actor is authorized to view.
Room View Page	User's identity Active Room	Used to determine the actor's credentials, which in turn specify what data this actor is authorized to view. The Room that is currently being viewed in detail.

Use Case UC-8: ManageRooms

Concept	**Attributes**	**Attribute Description**
Room Manager	User's identity Active Room	Used to determine the actor's credentials, which in turn specify what data this actor is authorized to view. The Room that is currently being viewed in detail, if in Room View.

Traceability Matrix

Use Case	PW	Controller-UC-4	Interface Page-UC-4	Database Connection-UC-4	Login Page	Dashboard Page	Room View Page	Active Room	Controller-UC-8	Interface Page-UC-8	Database Connection-UC-8	Room Manager	Show User
1	21												
2	8												
3	24												
4	27	X	X	X	X	X	X	X					
5	3												
6	9												
7	15												
8	10								X	X	X	X	X
9	4												
10	4												

System Operations Contracts

Operation	SendInfo
Preconditions	System is already active and running in the background
Postconditions	To receive information from the arduino and display to the user

Operation	ReceiveInfo
Preconditions	Arduino is connected and has information to send
Postconditions	Web app receives arduino information

Operation	ShowUser
Preconditions	User has navigated to a new page on the web app.
Postconditions	The page requested by the user will be displayed.

Operation	ManageRooms
Preconditions	User is in Dashboard if adding a new Room; if removing or modifying a Room, there is a Room being targeted in Room View.
Postconditions	The requested action on a Room will be completed.

Mathematical Model

None used.

Project Size Estimation

Environmental Complexity Factor (ECF)
Assume:
ECF = **1**.

Unadjusted Use Case Weight (UUCW)

Use Case	Complexity Classification	Weight	Justification	No. of Transactions
1: Login	Simple	5	User inputs a username and password; the system validates this against a database and responds with whether the credentials are valid (1 transaction).	1
2: Send Info	Average	10	The Hygrometer and Thermostat feed data into the web app through the Arduino (two transactions). The web app updates its information and pages, according to its Mode (2 transactions).	4
3: Receive Info	Simple	5	The web app requests information from the Arduino, which sends the web app its latest information via SendInfo (1 transaction).	1
4: Show User	Simple	5	The User navigates to a new page, and the web app requests that page to be displayed (1 transaction).	1
5: Show Dash-board	Simple	5	The User logs in or navigates to the Dashboard; the system fetches their Room data from the database, and displays the Dashboard to the User (1 transaction).	1
6: Show Room View	Simple	5	The User clicks on a specific Room; the system fetches their Room data for that Room from the database, and displays the Room View to the User (1 transaction).	1

7: Show Login	Simple	5	The User logs out or navigates to the web app for the first time; the system displays the Login Page (1 transaction).	1
8: Manage Rooms	Complex	15	The User requests to add a new Room; they are prompted to specify parameters, and respond by supplying requested information; the system sends this to the database and requests the Room View for this Room (3 transactions). The User requests to remove a Room; they are prompted for confirmation, and respond by confirming or denying; if confirmed, the system sends this to the database and requests the Dashboard View (3 transactions). The User requests to modify a Room; they are prompted for new parameters, and respond by supplying requested information; the system sends this to the database (2 transactions).	8
9: Forum Post	Average	10	The User writes a forum post and submits it; the post is sent to the database, and the forum is updated (1 transaction). The User requests to remove a forum post; they are prompted for confirmation, and respond by confirming or denying; if confirmed, the system sends this to the database (2 transactions). The User requests to modify a forum post; they are prompted to make their edits, and respond by making changes; the system sends this to the database (2 transactions).	5

10: Switch Mode	Simple	5	The User requests a Mode switch, and the system responds by toggling their mode from Greenhouse to Residence, or vice versa (1 transaction).	1

Total Weight (UUCW) = (7 Simple UCs • 5) + (2 Average UCs • 10) + (1 Complex UC • 15)
UUCW = **70**

Unadjusted Actor Weight (UAW)

Actor	Complexity Classification	Weight	Justification
User	Complex	3	Human actor using a GUI application interface.
Web App	Simple	1	Connects to both the Arduino setup and the database using well-defined API.
Database	Average	2	External system that interacts with the system using standard communication protocols.
Hygrometer	Average	2	External system that interacts with the system using standard communication protocols.
Thermostat	Average	2	External system that interacts with the system using standard communication protocols.
Arduino	Simple	1	External system that interacts with the system using a well-defined API.

Total Weight (UAW) = (2 Simple Actors • 1) + (3 Average Actors • 2) + (1 Complex Actor • 3)
UAW = **11**

Technical Complexity Factor (TCF)

Factor	Description	Weight	Score
T1	Distributed System	2.0	3
T2	Response time/performance objectives	1.0	5
T3	End-user efficiency	1.0	5

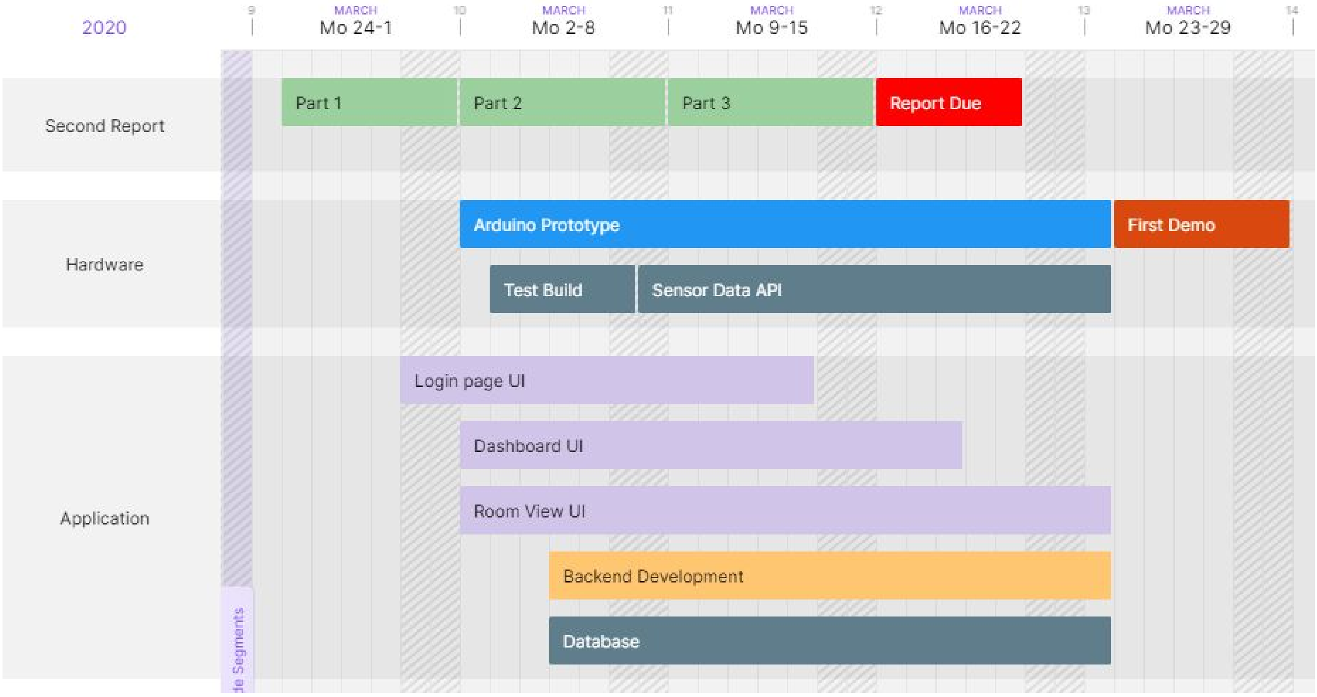
T4	Internal processing complexity	1.0	3
T5	Code reusability	1.0	3
T6	Easy to install	0.5	2
T7	Easy to use	0.5	5
T8	Portability to other platforms	2.0	4
T9	System maintenance	1.0	4
T10	Concurrent/parallel processing	1.0	0
T11	Security features	1.0	3
T12	Access for third parties	1.0	0
T13	End user training	1.0	0

Technical Factor (TF) = (∑each score • each weight) = 40.5
TCF = 0.6 + (TF / 100) = **1.005**

Use Case Points (UCP)
UCP = (UUCW + UAW) • TCF • ECF
UCP = (70 + 11) • 1.005 • 1 = **81.405**

Plan of Work

As soon as we finish this report, we plan on starting our second report as well as slowly transitioning to our UI development. At the same time, we'll have our hardware team begin testing some prototypes for our hygrometer and thermometer sensors. They will also need to implement an API for the hardware to communicate with our web app. Additionally, our backend development team will be in constant communication with our UI team in order to implement a robust framework for our application's features.



<u>Product Ownership:</u>

- UI Development: Hrishit and Gabe
- Hardware Team: Omar and Olek
- Backend Development: Nathan and Greg
- Database: Alex and Jake S.
- Model Calculations: Yanbo and Jake H.

References

Lucidchart.com. (2020). [online] Available at: https://www.lucidchart.com/pages/uml-use-case-diagram [Accessed 17 Feb. 2020].

Used for extra information about Use Cases.

Ambius.com. (2020). How Does Humidity Affect Plants? | Ambius US. [online] Available at: https://www.ambius.com/indoor-plants/humidity-affects-plants/ [Accessed 17 Feb. 2020].

Used for information on how to increase humidity.

Maker Pro. (2020). How to Make an Arduino Thermometer and Hygrometer With OLED Display | Arduino. [online] Available at: https://maker.pro/arduino/projects/arduino-thermometer-hygrometer-oled-display [Accessed 17 Feb. 2020].

Used for instruction on how to connect a hygrometer and thermometer to an arduino.

Greenhouse Product News. (2020). Managing Humidity in the Greenhouse - Greenhouse Product News. [online] Available at: https://gpnmag.com/article/managing-humidity-in-the-greenhouse/ [Accessed 17 Feb. 2020].

Contains a lot of efficiency values, highlighting how heat and vent systems are affected by humidity.

Center for Agriculture, Food and the Environment. (2020). Reducing Humidity in the Greenhouse. [online] Available at: https://ag.umass.edu/greenhouse-floriculture/fact-sheets/reducing-humidity-in-greenhouse [Accessed 17 Feb. 2020].

Used for information on how to reduce humidity and current practices used.

Storey, A. (2020). The True Costs of Greenhouse Humidity Control - Upstart University. [online] Upstart University. Available at: https://university.upstartfarmers.com/blog/greenhouse-humidity-control [Accessed 17 Feb. 2020].

Used for information on humidity and its effects and costs to greenhouses.

Online.visual-paradigm.com. (2020). Visual Paradigm Online Diagrams. [online] Available at: https://online.visual-paradigm.com/drive/#diagramlist:proj=0&new=SequenceDiagram [Accessed 17 Feb. 2020].

Used for creation of UML diagrams and information regarding how to draft them.