

Smart Irrigation Controller

Man Ha, ham@oregonstate.edu

Maxwell Okazaki, okazamax@oregonstate.edu

Joshua Barringer, barrinjo@oregonstate.edu

Table of Contents

Design Impact Statement	3
Public Health, Safety, and Welfare Impacts	3
Cultural and Social Impacts	3
Environmental Impacts	3
Economic Factors	3
Design Impact Conclusion	4
Project Timeline	4
Scope and Engineering Requirements Summary	9
Risk Register	13
Future Recommendations	15
References	17

Design Impact Statement

Public Health, Safety, and Welfare Impacts

The Smart Irrigation project aims to make home gardening an easier task for both beginners and those with experience. An increase in home gardening will increase access to fresh fruits and vegetables, promoting healthier eating as well as a healthier lifestyle since gardening provides physical activity to its participants [1]. On another note, this project features a database to store personal user information, including name, email address, location, and more. By storing this information online, there is always a possibility of cyberattacks and having user data stolen. Furthermore, users have little control over their personal data and how it is used in most current online settings [2]. A smart and efficient water irrigation system can help reduce the water waste. A steady supply of food and filtered water not only help reduce the hunger in these areas but also reduce the chance of waterborne disease.

Cultural and Social Impacts

The Smart Irrigation system in its current form is heavily targeted at the home gardener, which isn't inherently bad, but there is significant literature supporting urban agriculture and its benefits for building community, providing healthy food and recreation, beautifying neighborhoods, and making productive use of vacant land [3]. The act of gathering as a community to garden and work together has benefited many societies in the past. To best build up communities, the Smart Irrigation system can better support communities and gardens (over just individuals) through cooperative efforts to increase the amount of community gardens.

Environmental Impacts

One of the biggest draws to the Smart Irrigation project is its water saving functionality. Depending on the region, homeowners use 30-60% of their water outdoors and 50% of that is wasted, in part, due to overwatering. Furthermore, it has been shown that certain smart irrigation systems can reduce water use by 15%, which saves a significant amount per year [4]. This project uses a similar system as those used in the studies, where local weather data is used to determine when and how much to water. Creating an internet-connected garden management system would likely have an undetectable amount of carbon emissions, but it contributes to a larger pattern of increasing internet and technology usage that is creating significant carbon emissions [5]. Most of the environmental impact comes from the use of irrigation systems as the water moves from lakes, rivers, and underground sources. Irrigation removes a large volume of water from a given area and transports it to rural or landscaped areas. Other uses such as solar plan implementation and additional water filtration system will introduce more human footprint in surrounding natural areas. The irrigation system needs to research for a better and more efficient way to transport water as these problems can be solved by using better materials, further care, and maintenance to reduce the risk of pollution and minimize ecological disruption [6].

Economic Factors

The modernization of farming technology has the potential to greatly increase the economic yield of future farms and greenhouses. Currently, this technology is being applied in greenhouses, where growth parameters can be controlled more consistently. Researchers have started developing neural networks to predict and optimize plant growth based on these parameters, which might lead to more optimized food production [7][8].

Design Impact Conclusion

Many of the topics discussed in this impact assessment lead us to believe that this project will ultimately impact these areas in a mostly positive way. Most of the impacts in the economic and environmental areas can only be truly realized at scale, if the Smart Irrigation system grows and finds its way into many homes (and maybe eventually commercially). But on the individual level, one person's overall wellness can be greatly impacted by the adoption of our system and implementation of a more active and healthy lifestyle that gardening will afford.

Project Timeline

Table 1: A table showing the initial project timeline that we created. Updated: 11/12/2020.

Week	Activities	Status
(week 3)	Team formed, meet with project sponsor, engineering requirements draft.	Completed
(week 4)	Communication evaluation meeting, risk register (individually), research individually.	Completed
(week 5)	Writing workshop.	Completed
(week 6)	Draft block diagram, bi-weekly progress video.	Completed
(week 7)	Update sponsor, instructor system architecture meeting, project charter.	Completed
(week 8)	Teamwork reflection video, bi-weekly progress video.	Completed
(week 9)	Engineering requirements, block diagram.	Completed
(week 10)	Research implication report, project partner update, bi-weekly progress video, block validation(s).	Completed
(fall finals week)	Prototype demonstrations	Completed
(week 13)	Majority of blocks of the system are done, Block validation 1. Research Implication Report draft.	Completed
(week 15)	All blocks of system design done. Block Check off 1.	Completed
(week 16)	Block Validation 2	Completed
(week 17)	Block Check off 2	Completed
(week 19)	RIR Final Draft	Completed

(week 20)	Block Check off 3 Finalize project blocks.	Completed
(week 21)	Team meeting and check for individual weekly availability.	Completed
(week 22)	Individual Elevator Speech.	Completed
(week 23)	Complete 3 requirements and ready for Initial System Check off.	Completed
(week 24)	Initial System Check off.	Completed
(week 25)	Project closeout drafted.	Completed
(week 26)	Finalize the system, combine the database, API and Web Interface all together. Fix all of the bugs. Filming all of the Requirement for Final System Check Off.	Completed
(week 27)	Final System Checkoff	Completed
(week 28)	Project Close Out. Showcase Assignment. Plan meeting with sponsor.	Completed on 5/20

All major assignments were completed on time.

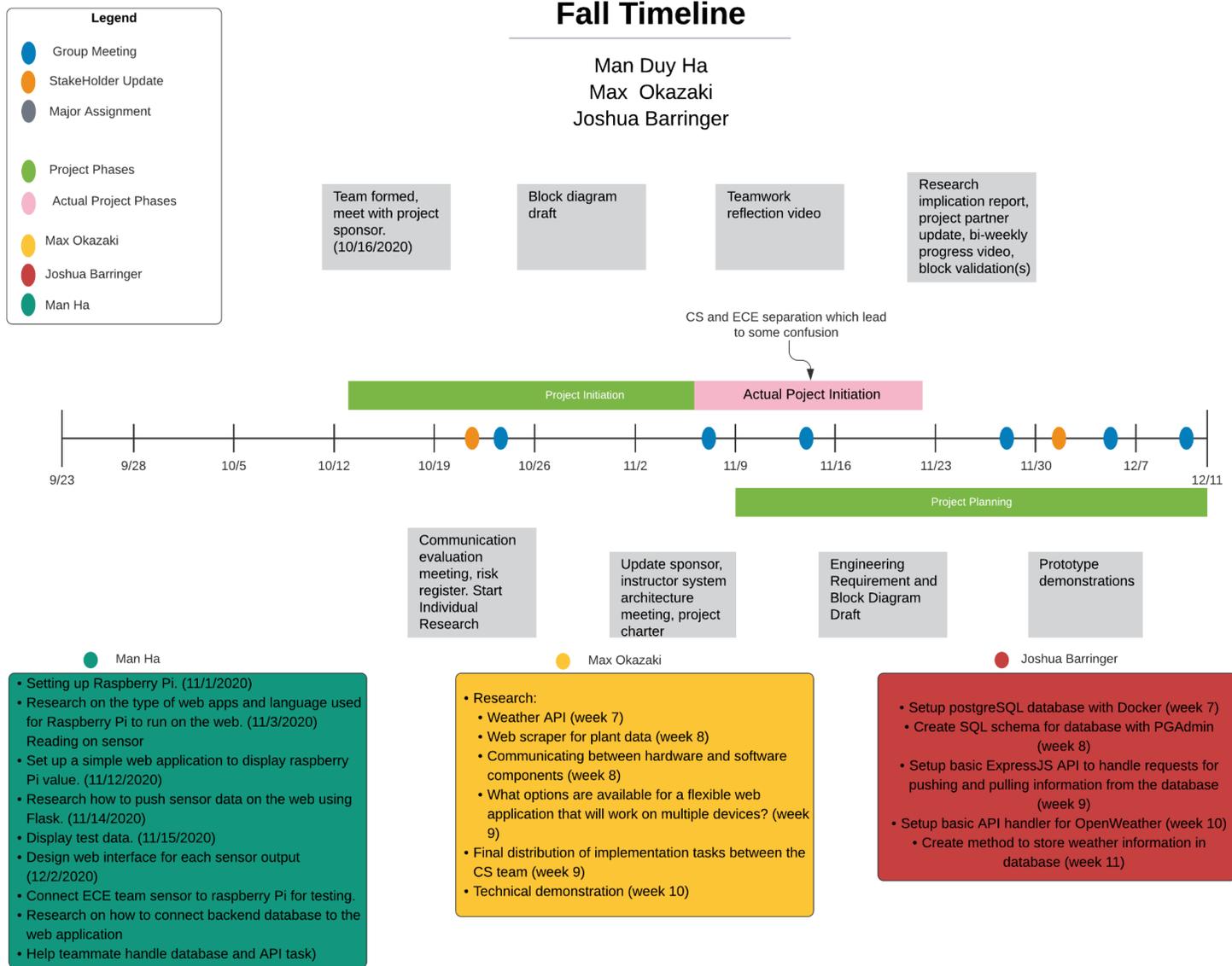


Figure 1: Fall Timeline

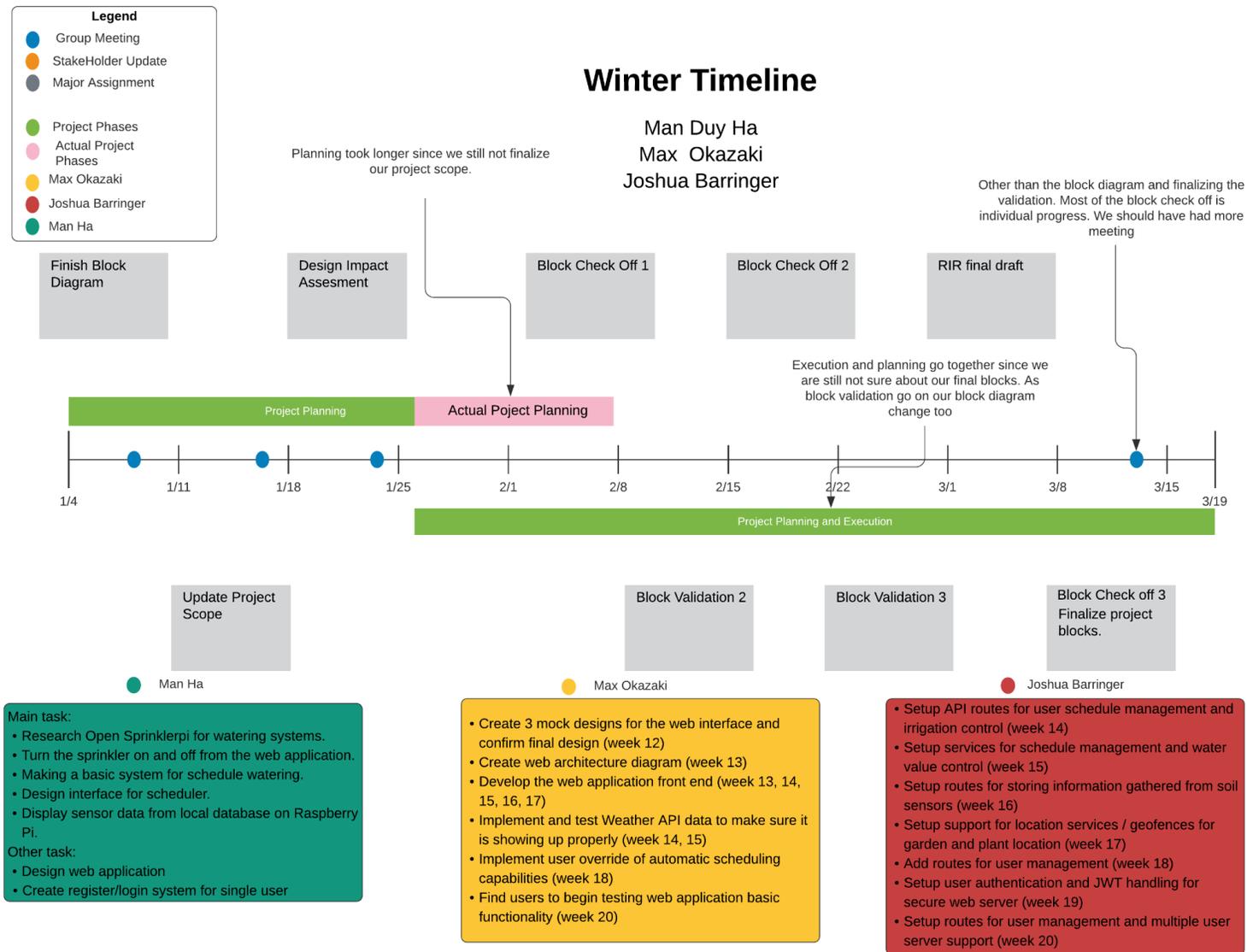


Figure 2: Winter Timeline

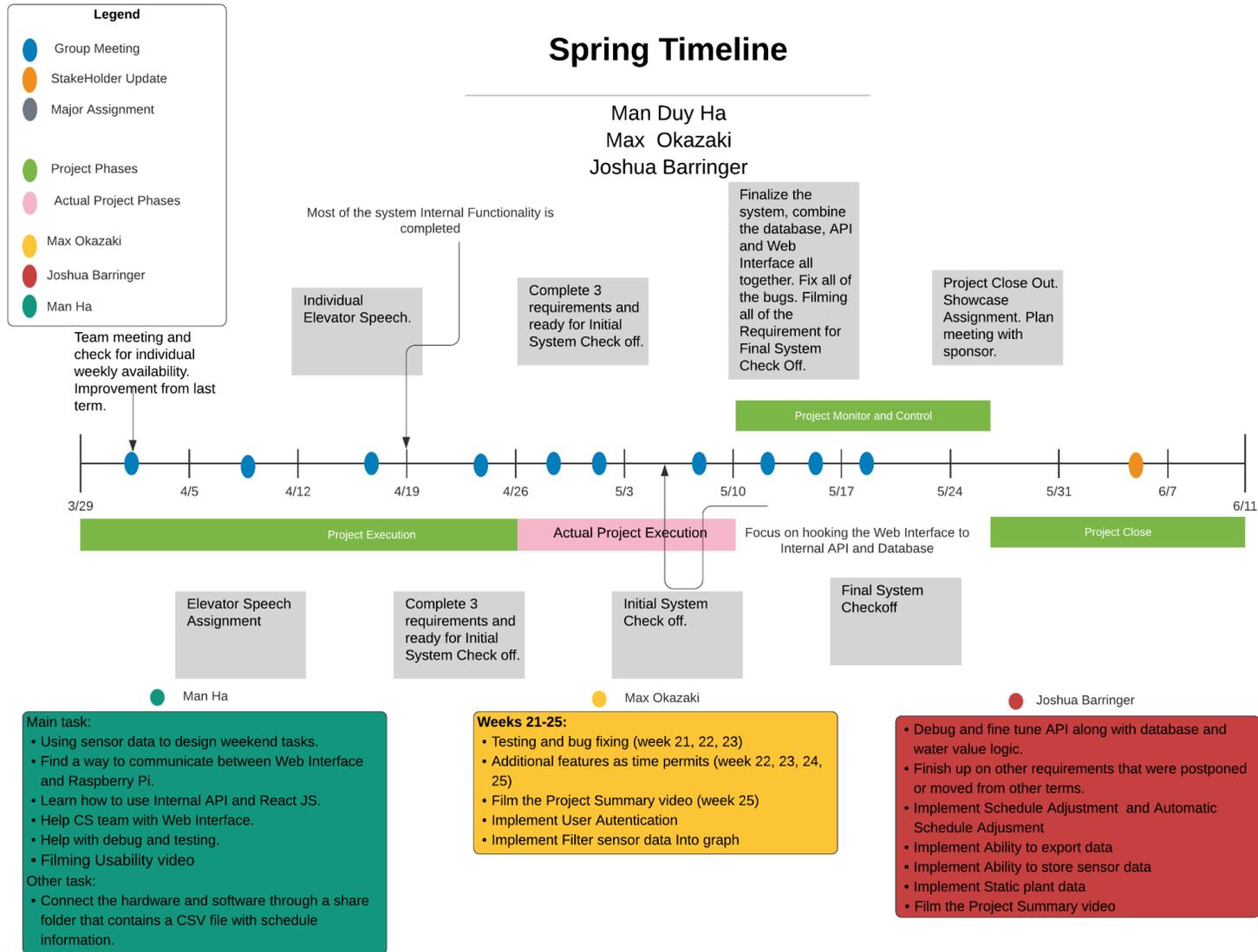


Figure 3: Spring Timeline

Scope and Engineering Requirements Summary

Table 2: Scope and engineering requirements.

Requirement	Verification Method	Testing Process	Testing Pass Condition
<p>1. Schedule Adjustment</p> <p>Project Partner Requirement: PP wants to be able to override the automatically created schedule as needed.</p> <p>Engineering Requirement: The system allows users to customize their watering schedule.</p> <p>Evidence Link:</p>	Demonstration	<ol style="list-style-type: none"> 1. User clicks on "Schedule" in the sidebar. 2. User adds in custom schedule values and clicks the "Submit" button. 3. Visually inspect the user's page for the new schedule. 	Schedule information can be shown before and after user interaction with clear changes to user schedules.
<p>2. Usability</p> <p>Project Partner Requirement: PP wants a polished interface that is easy to use.</p> <p>Engineering Requirement: The system interface will be stated as easy to use by 9 out of 10 people.</p> <p>Evidence Link:</p>	Test	<ol style="list-style-type: none"> 1. Testing participants will be given 15 minutes to use the interface with instructions to adjust a watering schedule. 2. Testing participant will be given a one-question survey with 4 options: <ol style="list-style-type: none"> 2a. The interface was very easy to use. 2b. The interface was easy to use. 2c. The interface was confusing to use. 2d. The interface was unusable. 3. Repeat for 10 users. 	9 out of 10 users rated the interface as either "easy" or "very easy" to use.

<p>3. Ability to Store Sensor Data</p> <p>Project Partner Requirement: Project partner wants ability to view aggregated data from available sensors over a period of time.</p> <p>Engineering Requirement: The system displays a table with sensor data which will include temperature, humidity, light, and soil moisture value.</p> <p>Evidence Link: https://youtu.be/XEN6EXvpZbw</p>	<p>Demonstration</p>	<ol style="list-style-type: none"> 1. User will click on "Dashboard" to access a table that displays sensor data. 2. Visually inspect the "Sensor Data" table for temperature, humidity, light, and soil moisture values. 	<p>The system shows newly inserted sensor data (temperature, humidity, light, and soil moisture) in the Web Interface.</p>
<p>4. Ability to Export Schedule Data</p> <p>Project Partner Requirement: Project partner wants the system to automatically activate/deactivate sprinklers according to the user defined schedule.</p> <p>Engineering Requirement: The system exports data for controlling sprinklers based on user adjustments.</p> <p>Evidence Link:</p>	<p>Demonstration</p>	<ol style="list-style-type: none"> 1. User clicks on "Schedule" in the sidebar to create a schedule change. 2. User adds in custom schedule values and clicks the "Submit" button. 3. Open the downloaded CSV file and visually inspect for the updated schedule changes. 	<p>If a CSV file can be opened that reflects the user-updated schedule.</p>

<p>5. Secure Personal Data</p> <p>Project Partner Requirement: Project partner suggested user management in the system so that many people could use the system</p> <p>Engineering Requirement: The system will only allow a user to see their own personal data based on their user id.</p> <p>Evidence Link: https://media.oregonstate.edu/media/t/1_ygshap3w</p>	<p>Demonstration</p>	<ol style="list-style-type: none"> 1. User logs in by clicking "Log in" and inputting their username/password. 2. User will navigate to their profile by clicking "Profile" 3. Visually inspect the information on the Profile page to match the user-provided information for their personal information. 4. Have another user repeat steps 1-3 to show that their personal information is different. 	<p>A user will only be able to view data associated with their account.</p>
<p>6. Filter Data into Graphs</p> <p>Project Partner Requirement: PP wants to be able to effectively view previous soil data (possibly in graph form rather than just tables)</p> <p>Engineering Requirement: The system can filter and display sensor data associated with a user only when that user is logged in.</p> <p>Evidence Link:</p>	<p>Demonstration</p>	<ol style="list-style-type: none"> 1. User will click on "Historical Data" to view their user-specific historical data. 2. User will select a date range and/or sensor type to be displayed. 3. Visually inspect that the graph shows the correctly filtered data. 4. Have another user login and repeat steps 1-3, selecting the same filters, to show that their sensor data is different. 	<p>Will display selected sensor data associated with a user's ID through the system interface.</p>
<p>7. Automatic Schedule Adjustments</p>	<p>Demonstration</p>	<ol style="list-style-type: none"> 1. User logs in by clicking "Log in" and inputting their username/password. 2. User will navigate to their schedule 	<p>If the expected precipitation changes, the schedule data should be adjusted accordingly</p>

<p>Project Partner Requirement: Project partner wants a system to adjust scheduling based on live weather information.</p> <p>Engineering Requirement: The system will automatically make adjustments to the watering schedule based on the next day's precipitation forecast.</p> <p>Evidence Link:</p>		<p>by clicking "Schedule"</p> <ol style="list-style-type: none"> 3. Visually inspect the current schedule for the next day. 4. Manually increase the expected precipitation for the next day. 5. User will refresh their page and visually inspect that the schedule has changed to reflect less watering for the next day. 6. Repeat steps 1-5 but decrease the expected precipitation and visually inspect for more watering for the next day. 	<p>(decrease in watering time if an increase in expected precipitation and an increase in watering time if a decrease in expected precipitation)</p>
<p>8. System Has Ideal Plant Watering Information</p> <p>Project Partner Requirement: PP suggested having certain plant data (like ideal watering conditions) available to users.</p> <p>Engineering Requirement: The system will suggest watering times based on a plant type selection menu.</p> <p>Evidence Link:</p>	<p>Demonstration</p>	<ol style="list-style-type: none"> 1. User will navigate to the "Plant Information" page using the navigation bar. 2. User will select a plant from the plant menu. 3. The page will display the selected plant's information (suggested watering levels, description). 	<p>The system will display a description, suggested watering levels, and name based on the selected plant.</p>

Risk Register

When the risks associated with our project were considered at the very beginning, before any work had been done, the table below is what we came up with. This was created when both the hardware and software teams were working as one, so 3 out of the 5 risks we came up with pertained to hardware, which we will not directly speak to in this project closeout.

As the project moved through its lifecycle and certain plans were changed due to time and other restraints, the fourth risk of the app not working on all devices wasn't necessarily encountered as we instead decided to only focus on a web application. The main reason for switching to focus on just one platform is that creating a great app that works seamlessly with the various platforms takes significantly more time and energy. Our team decided to alter the scope of this project to only focus on a web application.

Our team did not plan for the hardware team to lose a member and so when learning about that team being one member down, we decided to work in a more siloed manner and were not able to actually connect the software and hardware. In theory, since our main points of contact to the hardware team is through the transfer of data (a watering schedule and soil sensor data), our systems should be able to work together since the software team was successfully able to demonstrate a CSV file with the watering schedule being exported and the reading of sensor data in the database.

A great lesson we learned from the mixup between the software and hardware teams is to define and design where the teams work will intersect early on and stay in communication throughout the project. Meet more frequently and just stay up to date with how each team is doing, especially if both teams are working on the same ultimate project.

Table 3: The risk register that was created back in November, 2020 when the ECE and CS teams were still unified.

Risk ID	Risk Description	Risk Category (Cost, Technical, Communication)	Risk probability (0-100%)	Risk Impact (High, Medium, Low)	Performance Indicator	Responsible party (Man Ha, Joshua, Max)	Action Plan (Avoid, Retain, Reduce, Transfer)
R1	Product too expensive	Cost	30%	M	Price	Caleb	Retain
R2	GPS not precise enough	Technical	20%	H	Accurate down to 2.5m	Trenton	Retain
R3	Soil probe does not interface with the app	Technical	15%	H	The app receives data from probe	Trenton/Man	Reduce
R4	App doesn't work on all devices	Technical	35%	M	If it works on iOS and Android and PC	Max/Josh	Retain
R5	Database will not support multiple users	Technical	20%	M	The database has more than one user	Josh	Reduce

Future Recommendations

Table 4: Our future technical and project management recommendations.

Recommendation	Reason for Recommendation	Where to Start
<p>Make the connection between the hardware and software team wireless.</p>	<p>In the current system, we have a CSV file that holds a watering schedule that can be read by the hardware to control the irrigation system. If there is no wireless connection, the user would have to manually transfer this data to the hardware and that leads to a product that will not be used in today's connected world. To also fulfil a feature where we wanted to be able to instantaneously control the irrigation system, there will need to be some sort of wireless connection.</p>	<p>Work with the hardware team to figure out a wireless solution so that data and information can be sent between the two systems. We looked into Wi-Fi options for the Raspberry Pi.</p>
<p>Decide early on the project goal and figure out how the final product will look like.</p>	<p>Our project planning took longer than we expected since we tried to figure out all sorts of possibilities, the things we can do for our final product. Not until block validation came up that our final block diagram was finished. Furthermore, there are some confusions related to hardware-software integration and assigning the role for block check off.</p>	<p>Clearly laid out what we want to achieve and learn, what kind of features that are possible to implement in the span of two month from Winter to Spring. Spend winter break time to do research on what each member wants to do for block validation and check off.</p>
<p>Figure out each teammate's time availability early for team meetings.</p>	<p>During the Winter term we meet less because we didn't figure out each other's time availability, which led to some confusion related to major work assignments and assigning tasks.</p>	<p>Make an early meeting in the first week of the term and make a table to figure out each team member's availability, then use that table to make weekly meetings.</p>
<p>Meet with the project partner on a frequent basis.</p>	<p>Although we were tasked with sending bi-weekly updates early on in the year, these assignments stopped and weren't actually that realistic.</p>	<p>When initially meeting with the project partner, ask to set up a recurring meeting at a cadence that makes sense for everyone. We'd recommend meeting more</p>

	<p>Actually set up a meeting with the project partner so you can share updates AND get feedback. This will be critical to staying on track and the project partner should help you work out kinks.</p>	<p>early on in Fall when the scope of the project is being discussed and then maybe pick up the meetings again during Spring when things are really being worked on.</p>
<p>Early separation from ECE team for classwork and CS base project.</p>	<p>We split our teams into an ECE and CS team to focus development into our focus areas.</p>	<p>Working in ECE and CS teams can be a good way to have team members that are able to help with common problems among group members. Be wary of poor communication between teams. Staying up-to-date with your other team is a good way to reduce the amount of work necessary to complete the project.</p>
<p>Host the application publicly.</p>	<p>To actually turn this project into something of use, it has to be available for use out in the world. By hosting it publicly, people will be able to use it.</p>	<p>Look into various hosting websites and services. Consider the database, cost of services, and scalability.</p>
<p>Working in different development environments can create technical challenges. Make sure to fix this early to start initial and final project development.</p>	<p>Remote development introduces new problems to code-based projects. Everyone's development environment is different, and different software/libraries are not supported on different environments. This can introduce problems later into development that should be identified at an early stage</p>	<p>Make sure team members are fluent in standard version control management. New code should be written on new branches and merged in on completion. Meet with your team early and understand what each team member's development environments are to ensure that the languages and packages being used are compatible with all team members.</p>
<p>Make the app experience more seamless, fix and improve UI design.</p>	<p>In a time-crunch, visual design of a product gets pushed back. When there are close deadlines, the final product can work but look rough.</p>	<p>Make sure to get early development code done early so there is ample time for bug testing and visual improvements. Creating CSS and animations for web apps can be a more time-consuming process than one expects.</p>

References

- [1] R. Lal, "Home gardening and urban agriculture for advancing food and nutritional security in response to the COVID-19 pandemic," Food Security. Accessed: April 16, 2021. [Online] Available: <https://doi.org/10.1007/s12571-020-01058-3>
- [2] Kelly D. Martin and Patrick E. Murphy. "The role of data privacy in marketing". en. In: Journal of the Academy of Marketing Science 45.2 (Mar. 2017), pp. 135–155. ISSN: 0092-0703, 1552-7824. DOI: 10.1007/s11747-016-0495-4. URL: <http://link.springer.com/10.1007/s11747-016-0495-4>.
- [3] Hodgson, Kimberley, Campbell, Marcia C., & Bailkey, Martin. "Urban agriculture: Growing healthy, sustainable places," American Planning Association. Accessed: April 16, 2021. [Online]
- [4] "WaterSense Summer Infographic," Environmental Protection Agency. Accessed: April 16, 2021. [Online] Available: <https://www.epa.gov/watersense/watersense-summer-infographic>
- [5] Burcu Ozcan and Nicholas Apergis. "The impact of internet use on air pollution: Evidence from emerging countries". en. In: Environmental Science and Pollution Research 25.5 (Feb. 2018), pp. 4174–4189. ISSN: 0944-1344, 1614-7499. DOI: 10.1007/s11356-017-0825-1. URL: <http://link.springer.com/10.1007/s11356-017-0825-1>.
- [6] Amber Pariona. "What is the environmental impact of irrigation?". WorldAtlas. April 25, 2017. [Online]. Accessed 4/13/2021.
Link: <https://www.worldatlas.com/articles/what-is-the-environmental-impact-ofirrigation.html#:~:text=Direct%20Environmental%20Impact%20Of%20Irrigation&text=Increased%20evaporation%20in%20irrigated%20areas%20can%20cause%20instability%20in%20the,levels%20in%20the%20surrounding%20atmosphere>.
- [7] Xing-Rong Fan et al. "A knowledge-and-data-driven modeling approach for simulating plant growth and the dynamics of CO₂/O₂ concentrations in a closed system of plants and humans by integrating mechanistic and empirical models". en. In: Computers and Electronics in Agriculture 148 (May 2018), pp. 280–290. ISSN: 01681699. DOI: 10.1016/j.compag.2018.03.006. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0168169917306531>.
- [8] Robert G. de Luna et al. "Tomato Growth Stage Monitoring for Smart Farm Using Deep Transfer Learning with Machine Learning-based Maturity Grading". In: AGRIVITA Journal of Agricultural Science 42.1 (Feb. 2020). ISSN: 01260537, 23026766. DOI: 10.17503/agrivita.v42i1.2499. URL: <https://agrivita.ub.ac.id/index.php/agrivita/article/view/2499>.