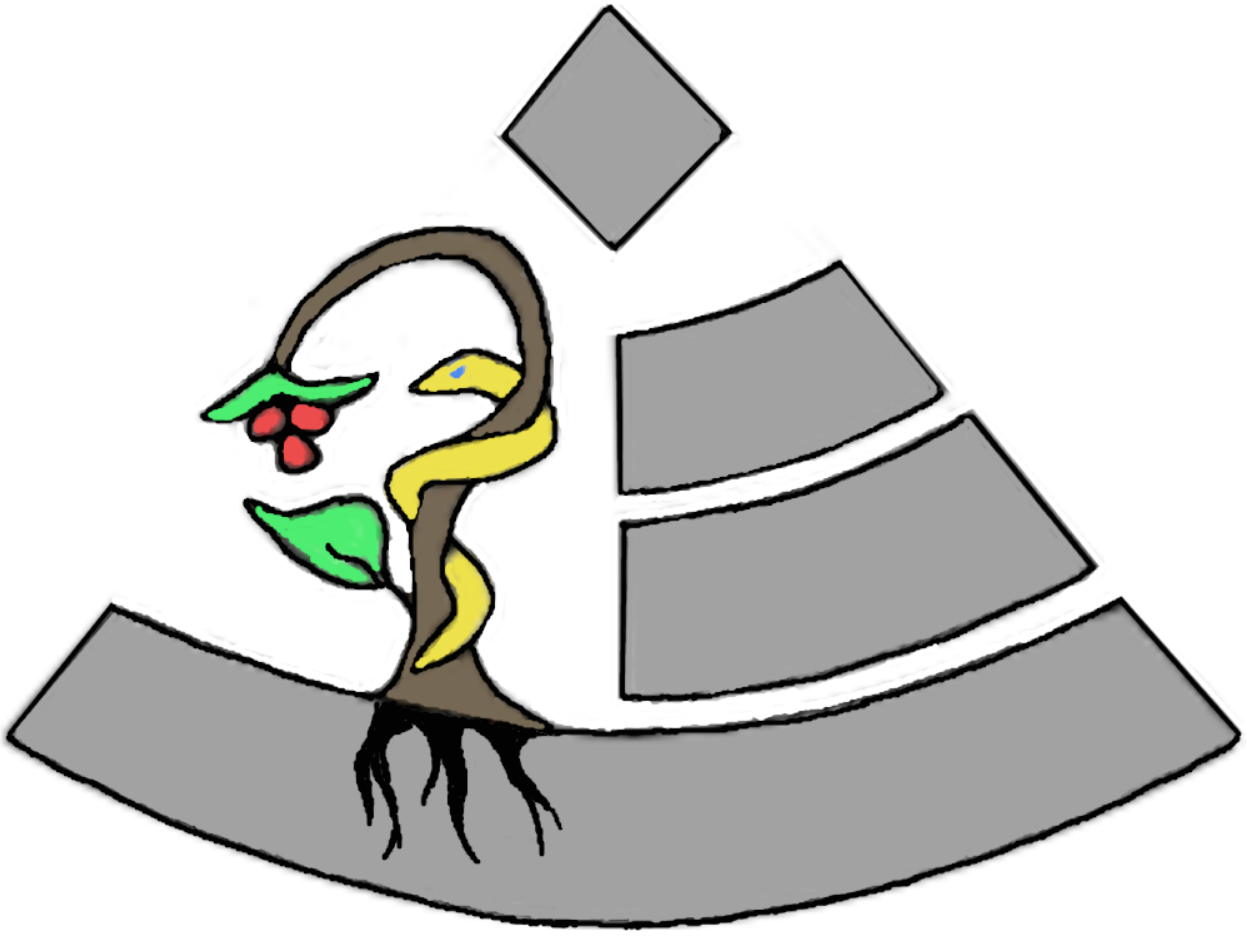
Greenwatch

Final Report

G. Mathers, D. Pollock, D. Portillo, K. Pulliam, E. Mondragon

4/10/2023



1 - Introduction

Greenwatch was built as a solution for monitoring environmental conditions in the MSU greenhouse. The system runs as a web server that can be accessed from anywhere using a modern browser, making it perfect for distributed use among the biology department. Greenwatch was intended to allow for more fruitful experimentation in the greenhouse as well as to produce a working relationship between the Computer Science and Biology departments to implement further solutions in the greenhouse as add-ons to Greenwatch. We used feature driven development as our SDLC of choice simply because we wanted to be able to add on features after the baseline was established, which allowed for us to add on several key features not originally mentioned in the requirements.

2 - Specification Phase

Our specifications were defined after the first initial meeting with Dr. Anthony Castilla. The team leader and hardware lead attended this meeting and found out the desired functionality within the greenhouse (well beyond what we could accomplish in a single semester). We were able to narrow down the specifications to simply record temperature/humidity to a database that would be accessible over the MSU public network. When we found out that this wasn't possible the scope of the project increased dramatically as we had to build our own virtual machine hosted in the cloud and produce a production workflow to allow for safety of the remote resources. With this being said, we delivered a product far beyond what was originally mentioned and added two units of measurement, light and pressure. Rendering the project a success.

3 - Planning Phase

The original design and architecture was produced by the team leader. This includes what technologies we were going to use, the fact that it was a website, and how we were going to bring everything together. It was decided to split the team into three groups. Front end, backend, and hardware all had individualized tasks in the plan that had due dates attached to them. After this was reviewed and discussed with the group, we were able to assign each major design document to each individual group. UML diagram to backend, swimlane diagram to frontend, and a general mapping of our hardware placement in the greenhouse. Though our plan was sound, we did not always stick to it. In some situations we were way ahead of the plan but in some we did not properly analyze the difficulty of the task and were put way behind. Our final sprint for the deployment phase took us probably 50 hours, it was expected to be only about 10 hours. Another lack of planning was on the part of the user interface. The team leader was not as familiar with the front end programming as the back end programming and

this showed greatly when we were in our last month of the project and all the user interface had was a login page and a semi functioning home page. Much more time would have been placed on the UI and deadlines would become non-negotiable.

4 - Implementation

4.1) Front-End

The process of developing the user interface was extremely slow at first. Those assigned to work on it had close to no knowledge in structuring HTML, CSS, and Javascript files together. There was a massive learning curve for the group when it came to understanding api calls and the various other modules used within the software. Members had to read through documentation before implementing new features. When features were developed, it would often undergo many more iterations of improvement visually and programmatically to ensure a smooth experience for the user.

There were many issues that arose as development on the user interface proceeded. One major issue at first was not having CORS, or Cross-Origin Resource Sharing, enabled on our flask application. It is a form of browser security that prevents us from making the requests we need to access our database. Thankfully it was fairly simple to fix as flask offers a way to enable CORS in your application.

4.2) Back-End

The implementation of the backend was done to schedule. A couple problems arose when connecting the front end to the back end. One such problem occurred when deleting a user. All messages that had that user as a relationship would set the user field to null. The user field was a non-nullable field so it would send DBAPI errors back detailing the issue. With this description we were able to find the error quickly and get it fixed. These changes were easy to make since we were using a simple SQLite database that we could delete whenever during development to have the tables and fields reset. This was very helpful because local development was simplified because we still had a SQL based database to connect the Python server to and test with, but in the end for production, everything ported straight in with no worries about PostgreSQL doing anything too much differently.

Overall, few errors arose until the end when we were deploying to production and variables being used across the application were statically set in various files. This resulted in mayhem. We converted all configuration details of the app to be editable from a single “.flaskenv” file. With all the configurable information across the application in one place it was super easy to switch from local development to remote production environments. Now, all files reference the .flaskenv for application level information.

4.3) Hardware

All hardware related tasks were done within a week of receiving the actual hardware. This allowed for great insight into the capabilities of the devices we purchased. Many problems that we faced on the hardware side were alleviated with diligent research into the devices documentation. This is all vividly detailed in our /Hardware folder.

5 - Testing

5.1) Front-End

The black-box and gray-box testing techniques were used when confirming all components of the user interface behaved correctly. As the user interface was being developed, each feature would then be immediately tested. Once the user interface was at a stable build, the entirety of it would then be tested in a single run through to ensure that each component worked correctly with other aspects of the software.

5.2) Back-End

White box testing was the main focus with the backend. Since all code was fully visible it was simply a matter of testing and ironing out any key details of the creation/updating/deletion/retrieval of resources from the database. Mainly, we used integration testing to ensure the python server was able to effectively communicate with the SQL database and also handled any exceptions thrown as a result of those communications.

5.3) Hardware

Black box testing was primarily used, as we had no prior knowledge to the inner workings of the sense-hat and raspberry PIs. We were able to test our way around the hardware to discover new features to implement early on in the project which helped us to go past the initial requirements.

6 - Defects

6.1) Front-End

There have been multiple failures discovered in the user interface while testing its behavior. When testing the application on different systems with varying screen sizes, it was discovered that certain aspects of the user interface components do not resize themselves to properly fit the screens proportions. This causes some components to appear larger or smaller than intended.

When testing the application with big data, it has been found that multiple loading points within the application take far too long to render user interface components on

screen. Also, when there is big data, such as there being tens of thousands of measurements being loaded from the database, the application begins to slow down significantly.

6.2) Connection of Front-End to Back-End (Javascript API Requests)

Due to the lack of knowledge on javascript web development at the start of this project, the structure of the project's source code was negatively impacted. While testing the application, it was discovered that there are far too many requests happening unnecessarily. For example, there were multiple occasions when measurements were pulled from the database when they only needed to be pulled once. This impacts the overall performance of the application by causing the application to respond slower than intended.

When handling the JWT's in the javascript requests, there would often be a failure that would not allow the user to perform any action in the user interface due to there being an unprocessable entity fault within the code for the javascript api calls.

6.3) Back-End

At the time of writing, there have been no issues discovered in the back-end during the testing process. Defects that could possibly arise are non-nullable fields becoming null because of cascading happening on the database when a deletion occurs. These cases have been tested though and should be fine now. Another is the server that the website is currently running on is very slow and has minimal resources. In the future, an upgrade may be necessary to have the application run effectively for multiple users at the same time.

7 - Conclusions

The following statements are conclusions of the project from each team member. They will include information about how they thought the project went, what was hard, what was fun, etc.

Garrett Mathers

This project was a great experience to dive into a very complex and broad problem. I learned a great deal about the planning/design phase that I never knew about. More commonly when I code I simply have an idea in my head about what I want

the code to do and accomplish. Had I done that with this project, it would never have been completed.

I also learned the strains of leadership and why it is good to be very diverse in your understanding of software technologies as a leader. Had I understood front-end technologies better we would've been able to implement a much more scalable and complete product than what we ended up with. I underestimated the time it would take to build my idea and as a result spent many restless nights trying to graft everything together in the end when in reality we should have already had it deployed with all intended functionality implemented. My lack of holding the front end team accountable also contributed to our tardiness. When the first deadline was not hit I was reluctant to impose stricter deadlines. It took until the third for me to finally realize how dire the situation really was. After a few more serious meetings and me being a little bit more explicative of each feature they too realized the situation they were now in.

Looking back, I should've had more individual time with each member to give a rundown of the codebase and what was in the works. I kept a lot of knowledge of the project in the code and my mind rather than actively sharing high level details with other group members.

Kolten Pulliam

Firstly, I'd like to mention that this entire project was a huge learning curve not just for myself, but for every group member as well. At the start, things seemed to be proceeding smoothly with the planning and scheduling. However, as soon as we got into implementing the user interface, we were basically walking into it blind. Having never built a web application before, I spent most of my time watching videos, reading documentation, and experimenting with prototypes as I stumbled my way through creating the user interface alongside Derrik. It didn't take long for me to realize how much I had underestimated the scale of this project.

I learned a great deal, especially right at the end during crunch time.

Edwin Mondragon

The greenhouse project that was presented to our group, not only was it a challenging one, but an extremely informative one. Whether it may be from the organization of the team, or the amount of understanding needed behind the topic at hand. Being in a situation where the group had to integrate into a completely new system is a problem that will more than likely happen to us as individuals in the real world. This new system was one that we as individuals came to an agreement on. It was through leadership and teamwork that led the team to success, but success does not always present itself as a straight path, there were several bumps and bruises. These bumps are what proved that no matter the individual, there is always a limit to what they can handle.

Being the main one working under the hardware side of things, may have seemed like an easy task, proved to be the latter. A firm understanding was needed so an explanation followed the questions of why certain components were used, and their intentions. While at first it may have seemed I was at my wits ends, my group proposed ideas that provided clarity and backs to rely on when something was not understood. Overall, the project was not only a fun one to solve but one that provided numerous experiences.

Derrik Pollock

Our greenhouse project went relatively smoothly. The group was well-organized and communication was effective. Each team member contributed to the project, and we were able to stay on track with deadlines. However, towards the end of the project, we had to rush the programming phase to meet our submission deadline. Despite the last-minute rush, we were able to complete the project and deliver a functional product.

In hindsight, there are several areas where we could have improved our project. One area is starting the coding portion earlier in the project timeline to allow more time for testing and debugging. Additionally, completing the user interface in the first half of the semester would have allowed for more time to focus on the back-end development. We could have also benefited from less meetings and more paired programming sessions to ensure that all members had a better understanding of both the front-end and back-end development processes. Overall, while we were successful in delivering the project, we could have improved our efficiency and workflow to make the project run more smoothly.

Daniel Portillo

I learned a great deal from this project in ways that I did and did not expect. I learned how to manage expectations and how aspects of the development can change throughout the entire year. Most importantly however I learned more about talking to and trusting teammates more as well as trying to adapt work during life changes. With the end of the project in sight I am more confident than ever that I can be more effective in groups in the future.

All the members did their tasks exceptionally well in order to create this piece of software in a relatively short amount of time. Though my part was relatively smaller than the others I was still able to to experience a substantial amount throughout the process. With the constant input and workflow from all members it provided an eye opening experience of some of the more nuanced aspects of SDLS and how members interacted with one another both in professional and friendly aspects. I believe it was due to the knowledge, character, and leadership of my teammates that allowed us to push through this project and its difficulties to achieve an effective solution to it.