Planter's Guide

An app to monitor your plant's health!

Christopher Butts

Kevin Eaton

Raymond Gee

Supervised by Dr. Badri Vellambi

Abstract

Planter's Guide is a mobile app that can monitor a plant's sunlight and moisture levels. The app communicates with a Raspberry Pi to send and receive info. The Raspberry Pi is connected to moisture and sunlight sensors as well as a SQL database to determine if a plant's moisture and sunlight levels are appropriate for the plant's species. The plant can also be automatically watered by a mechanism.

Project Description

Planters Guide is an Android application that can be utilized to monitor a plant's sunlight and moisture levels. The app will display the plant's current moisture and sunlight levels, as well as notifying the user if either of these levels are appropriate for the plant's species.

The application will communicate with a Raspberry Pi to get the plant's sunlight and moisture levels via two sensors that will be placed into the plant's soil. The Raspberry Pi will also be communicating with a remote SQL database which will store a variety of different plant species' appropriate water and sunlight thresholds. The Raspberry Pi will be sending data from both the SQL database and the sensors to the app in real time.

The application will also allow the user to turn on automatic watering if they are unable to reliably water their plant. The automatic watering will be done by a mechanism we develop that will communicate with the Raspberry Pi; if the water level gets too low below the water threshold, the machine will water the plant until it is comfortably above the threshold.

Project Focus:

- Android app development
- Back-end SQL database management
- Raspberry Pi Integration via sensors and wireless communication
- Automation

User Stories

Gale Cumbritch - Beginner Plant Mom

"As a plant owner, I want to be notified if my plant needs more or less sun and water. This way, I can make sure my plant is happy and healthy!"

Philip Dunks - The Busiest Businessman in Ohio

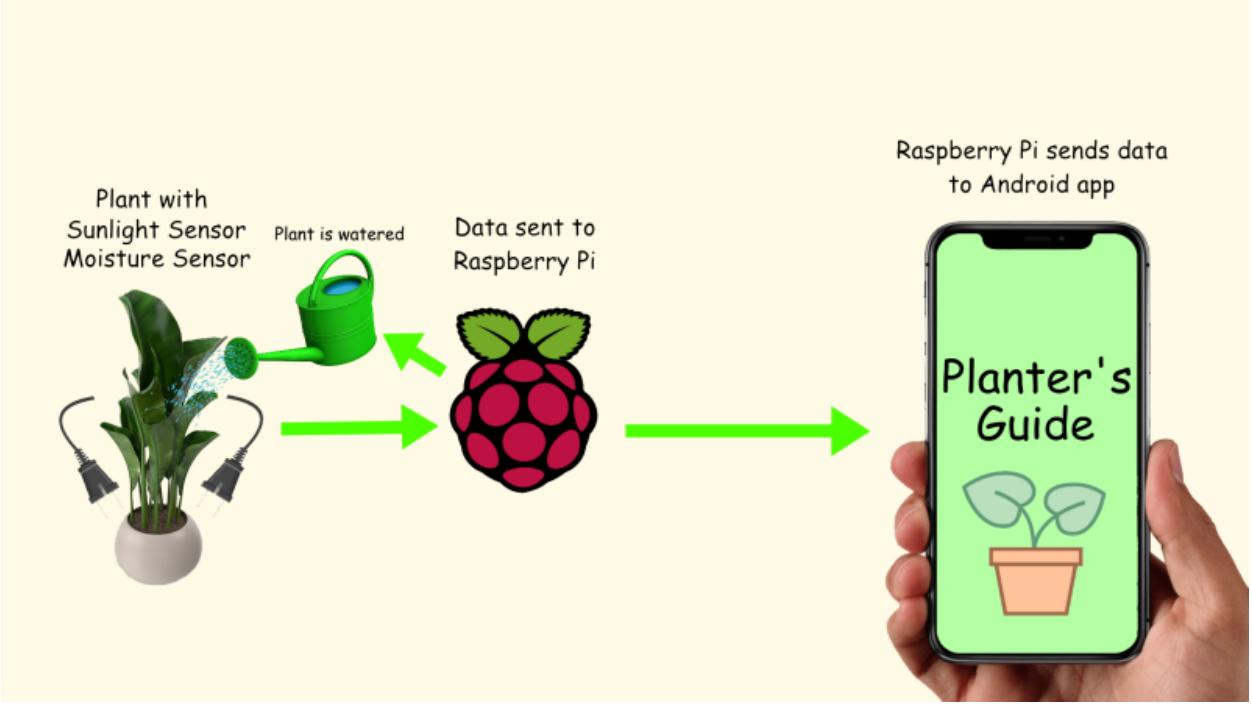
"As a forgetful person, I want an easy way that I can have my plant automatically watered when necessary. This way, my plant will not shrivel up and die if I forget to water it one day."

Alan Oldinsen - The Plant Whisperer

"As a plant enthusiast, I want to know the proper sun and moisture levels for my species of plant, so that I can adjust my plant's sunlight intake and my watering habits accordingly."

Design Diagrams

Level 0



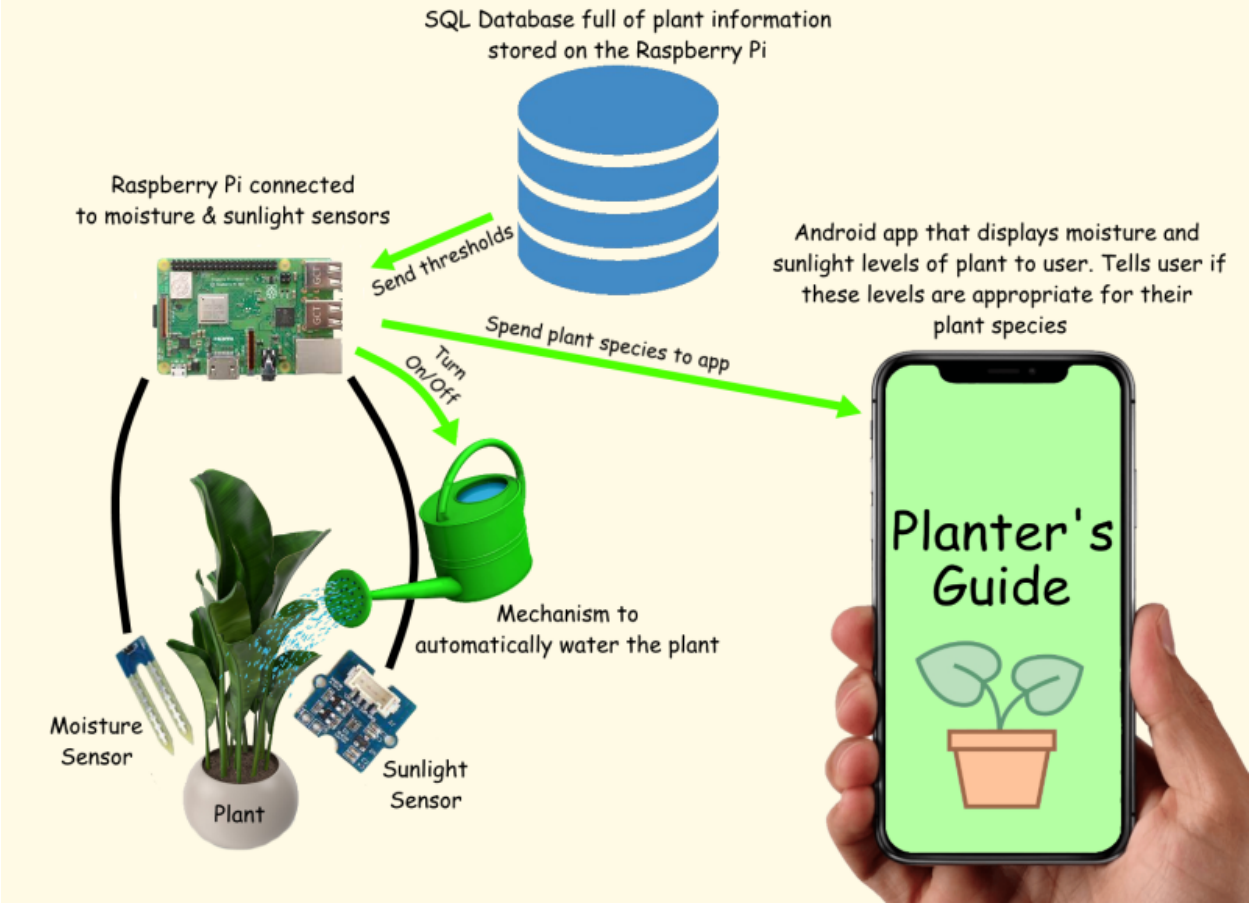
Plant
Connected to: Sunlight & moisture sensors, Raspberry Pi (via sensors)
Purpose: The main object of the product. The plant which you want to track the health of.

Raspberry Pi
Connected to: Sunlight & moisture sensors, automatic watering device, Android app
Purpose: The central intelligence hub. Does all communication and transferring of required data

Automatic Watering Device
Connected to: Raspberry Pi
Purpose: The device that will automatically water the plant when necessary.

Android App
Connected to: Raspberry Pi
Purpose: The app will report the health of the plant

Level 1



<u>Plant</u>
Connected to: Sunlight & moisture sensors, Raspberry Pi (via sensors)
Purpose: The main object of the product. The plant which you want to track the health of.

<u>Raspberry Pi:</u>
Connected to: Sunlight & moisture sensors, SQL database, automatic watering device, Android app
Purpose: The Raspberry Pi is the central communication hub. The Raspberry Pi sends the plant species data to the app, and gathers the health thresholds from the database to also send to the app. As well, the Raspberry Pi will send moisture or sunlight sensors' readings to the app and let the automatic watering mechanism know if it needs to be on or off.

SQL Database

Connected to: Raspberry Pi

Purpose: The SQL database will hold all the relevant plant information we need: plant name, proper sunlight level, and proper moisture level. The Raspberry Pi will send a select statement to the database, and the database will output the relevant line for the plant back to the Raspberry Pi.

Automatic Watering Device

Connected to: Raspberry Pi

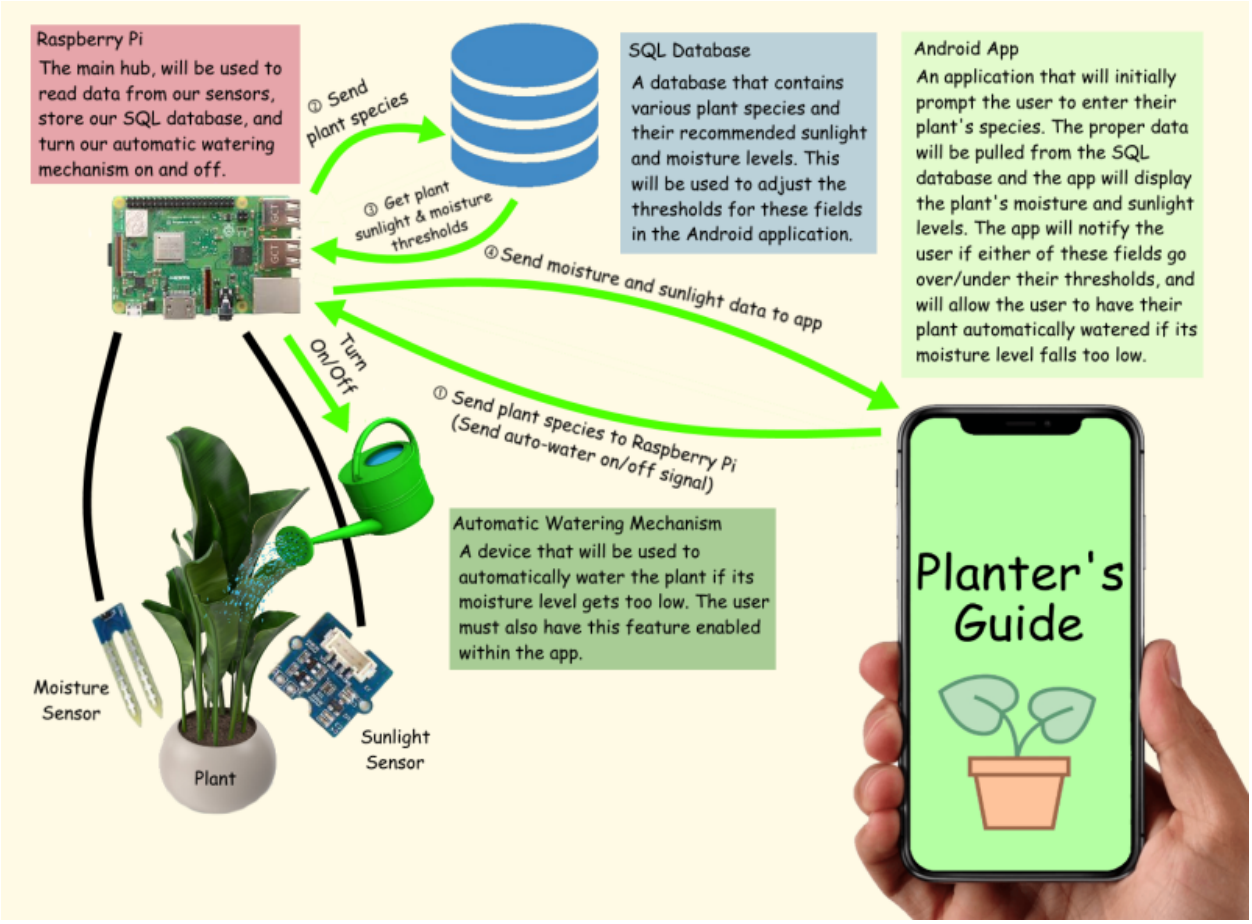
Purpose: The automatic watering device will water the device when necessary. The Raspberry Pi will receive a signal from the Raspberry Pi to denote when the watering device should be working or not.

Android App:

Connected to: Raspberry Pi

Purpose: The Android application will report a plant's moisture and sunlight levels, in real time via the moisture and sunlight sensors connected to the Raspberry Pi. The Raspberry Pi will send the information on the plant's species to the Android app, and the app will let the user know if the plant's current levels are good for the plant's species, and if not it will let the user know.

Level 2



<u>Plant</u>
Connected to: Sunlight & moisture sensors, Raspberry Pi (via sensors)
Purpose: The main object of the product. The plant which you want to track the health of.

<u>Raspberry Pi:</u>
Connected to: Sunlight & moisture sensors, SQL database, automatic watering device, Android app
Purpose: The Raspberry Pi is the central communication hub. The Raspberry Pi in a typical run situation will receive the plant species from the app, send a select statement using this data to the SQL database, receive the output, and return the relevant data to the app. As well, the Raspberry Pi will consistently send the moisture and sunlight sensors' readings to the app to update the displayed value, and the App will send a signal to the Raspberry Pi letting it know if the automatic watering mechanism should be turned on or off.

SQL Database
Connected to: Raspberry Pi
Purpose: The SQL database is a remote data center that will be a database of all the plants. The Raspberry Pi will send a select statement to the database, and the database will in turn return the appropriate output. In this case, the Raspberry Pi will send the plant species it wants the data for, and the SQL database will return that species' appropriate moisture and sunlight thresholds.

Automatic Watering Device
Connected to: Raspberry Pi
Purpose: The automated watering device is a mechanism that will automatically water the plant to an appropriate level above the threshold if the plant's moisture falls below it. The automatic watering will be controlled by the Raspberry Pi. The Android app will send a signal to the Raspberry Pi to let it know if the option of automatic watering should be on or off. If it's on, it will work as explained. If it's off, there will be no automatic watering.

Android App:
Connected to: Raspberry Pi
Purpose: The Android App is the UI and main interface an end-user will be interacting with. The app will display a plant's moisture and sunlight levels according to the sensors. To know if your plant is healthy, you will enter your plant's species, and this info will be sent to the Raspberry Pi, which will eventually return that plant's appropriate watering and sunlight thresholds. From here the app will notify the user if either the sunlight or moisture levels fall too low, and if automatic watering is activated, the plant will be automatically watered for the user.

Project Tasks and Timeline

Task List

Materials

- Purchase a plant
- Purchase sunlight & moisture sensors
- Draft a model for automatic watering mechanism
- Purchase components for automatic watering mechanism

SQL Database

- Research the best way for us to host our SQL database online
- Develop a SQL database that stores various plant species' names, recommended moisture levels, and recommended sunlight levels

- Determine and develop a course of action for if the user enters a name NOT in the database
- Develop security measures to prevent people from being able to freely edit the database

Data Transfer

- Develop a method to send user-input to Raspberry Pi
- Develop a method for sending a select statement based on user-input to SQL database
- Develop a method to have SQL output sent to Raspberry Pi
- Develop a method to have SQL output on Raspberry Pi sent to Android app
- Develop a method to send a signal to Raspberry Pi to let it know to automatically water the plant

App

- Develop the skeleton of the application. The skeleton should be a functioning app that will open all required screens
- Develop the rest of the components of the app that will communicate with the Raspberry Pi.
- Add notifications to the app that let the user know if the plants needs watered or more/less sunlight

Automated Watering

- Develop a machine that will automatically waters the plant if necessary
- Ensure automatic watering does not over water the plant

Timeline

Category	Task Name	Working Dates	Expected Completion Date
Materials	Purchase Plant	11/1/2021 - 12/11/2021	12/11/2021
Materials	Purchase Sensors	11/1/2021 - 12/11/2021	12/11/2021
Materials	Draft auto-watering mechanism	12/11/2021 - 1/10/2022	1/10/2022
SQL	Develop SQL database	12/11/2021 - 2/1/2022	2/1/2022
Materials	Purchase parts for auto-watering mechanism	1/10/2022 - 2/10/2022	2/10/2022
SQL	Research best way to host database	1/10/2022 - 1/17/2022	1/17/2022

Transfer	Develop Raspberry Pi to Android transfer	1/10/2022 - 1/24/2022	1/24/2022
Transfer	Develop Raspberry Pi to SQL transfer	1/10/2022 - 1/31/2022	1/31/2022
SQL	Develop SQL database troubleshooting	2/1/2022 - 3/1/2022	3/1/2022
Transfer	Develop SQL output to Raspberry Pi transfer	2/1/2022 - 2/14/2022	2/14/2022
Transfer	Develop Android to Raspberry Pi transfer	2/1/2022 - 2/20/2022	2/20/2022
App	Develop app skeleton	2/1/2022 - 2/8/2022	2/8/2022
App	Develop app communication components	2/1/2022 - 3/15/2022	3/15/2022
SQL	Develop SQL database security measure	3/1/2022 - 3/15/2022	3/15/2022
Transfer	Develop method to send auto-water signal to Raspberry Pi	3/1/2022 - 3/15/2022	3/15/2022
Automation	Develop auto-watering mechanism	3/1/2022 - 4/1/2022	4/1/2022
App	Develop notifications	3/10/2022 - 3/7/2022	3/7/2022
App	Create app UI	3/15/2022 - 4/1/2022	4/1/2022
Automation	Ensure auto-watering mechanism doesn’t overwater	4/1/2022 - 4/7/2022	4/7/2022
Milestone	Transfer functionality	1/10/2022 - 2/28/2022	2/28/2022
Milestone	SQL database	12/11/2021 - 3/15/2022	3/15/2022
Milestone	Android application	2/1/2022 - 4/1/2022	4/1/2022
Milestone	Auto-watering mechanism	3/1/2022 - 4/7/2022	4/7/2022

Effort Matrix

Task Name	Primary Team Member	Chris Effort	Kevin Effort	Raymond Effort
Purchase plant	Chris	100%	0%	0%
Purchase sensors	Kevin	10%	90%	0%
Draft auto-watering mechanism	Kevin	20%	60%	20%
Purchase parts for auto-watering mechanism	Kevin	0%	100%	0%
Research best way to host database	Raymond	0%	10%	90%
Develop SQL database	Raymond	15%	5%	80%
Develop SQL database troubleshooting	Raymond	0%	20%	80%
Develop SQL database security measures	Raymond	30%	30%	40%
Develop Android to Raspberry Pi transfer	Kevin	40%	60%	0%
Develop Raspberry Pi to SQL transfer	Raymond	5%	35%	60%
Develop SQL output to Raspberry Pi transfer	Raymond	0%	40%	60%
Develop Raspberry Pi to Android transfer	Chris	60%	40%	0%
Develop method to send auto-water signal to Raspberry Pi	Kevin	40%	60%	0%
Develop app skeleton	Chris	90%	5%	5%
Develop app communication components	Chris	70%	20%	10%
Develop notifications	Chris	100%	0%	0%
Create app UI	Chris	80%	10%	10%
Develop auto-watering mechanism	Kevin	10%	80%	10%
Ensure auto-watering mechanism doesn’t overwater	Kevin	20%	60%	20%

PowerPoint Presentation

The following is a link to the PowerPoint slides on our GitHub repository:
PowerPoint Presentation

Self-Assessment Essays

Christopher Butts

Our group's senior design project revolves around combining app development, back-end database management, and automation to create a comprehensive project that could be used to aid plant owners in their day to day lives. This project aims to display the health of the user's plant to the user in an easily digestible way. To be more specific on what our project is, we're going to be making an Android application that displays the moisture and sunlight levels for a plant. With this information, we can let the user know if these levels are appropriate and notify them if they need changed; in the case of moisture, we'll make a mechanism to automatically water the plant if its moisture level gets too low. This project will act as a combination of all the different facets of computer science myself and my group have learned over these last 5 years.

Each of the core concepts present in our project were initially presented to me in their own individualized experiences. Focusing on the concepts learned in school first, the following courses were contributing factors in my current knowledge:

ENED1020 – Engineering Foundations: Engineering Foundations is the course that gave me a small taste of mechanical engineering before I began focusing on software development. It was a while ago, but the course at least familiarized me with working with hardware.

CS4092 – Database Design and Development: Database Design & Development is the course that first introduced me to SQL, and all the capabilities the language has in storing data in a comprehensive database – useful information to know when we begin setting up our back-end plant database

Besides these courses, all my programming courses in general helped in establishing good programming habits and getting exposed to collaborative programming through group projects. Besides my classes though, the following co-op experiences helped contribute to my knowledge on how I will approach this project:

Self-Upskilling (Student): At the start of the COVID-19 pandemic, I lost the initial co-op opportunity and I had instead to partake in personal upskilling. While this experience was dull, it was not meaningless. During this upskilling, I spent a portion of my time learning how to program in Android – an obviously useful skill for our project.

Martin and Associates (CS Co-op): While most of my work did not focus on things applicable to this project, the language I worked in, ProvideX, had some integration to SQL, giving me a little more exposure to the language. I seldom wrote in SQL but having just a little more experience in the language will certainly come in handy down the line.

IT@UC (IT Consultant): My experience at IT@UC didn't teach me any applicable technical skills, but it taught me a lot about making information presentable to the common consumer. Interacting with people with no computer experience meant that you had to explain concepts to them in simple terms that they can understand. This is useful information to know when we move forward designing the UI for our Android app. I need to make sure that we keep the UI intuitive, and only display information that the user would find relevant.

Regarding motivation for this project, there are a few motivations. The main one is that I get to work on a comprehensive project with my friends! I knew I wanted to do something with them that we would have fun working on, so this idea was perfect – it is something with practical use for anyone with plants, but still comprehensive enough to be considered good for a capstone project. The second reason is that I wanted to buy a plant. I told myself I would purchase one before graduation, so this is the perfect opportunity.

Our initial approach was not too complex, we can go into this knowing we wanted to measure plant info and show it to the user. The difficult part was being sure it was comprehensive enough so that we weren't overwhelmed, but not to make it too easy of a project. We initially thought of doing the database, an app, more sensors, the light automation of watering, and image processing…but we slimmed it down to just using two sensors, the database, the app, and the automation. This is something that we could much more realistically reach, and it gives us a set goal to know when we are done. Once we get all these components finished, we'll work on any extra ideas as they come – additional notifications to the user, adding some security to our SQL database, etc. I know that our group will put everything we can into managing our time and making this project right, so I know that our final product will be something we can be proud of.

Raymond Gee

For my group's senior design project, we plan on using a device that monitors a plant's water level to send information to a user. This will involve our project needing an application to receive the data. Once the app receives the data it will use a data table to determine how much the plant should be watered and sent that information to the user. From my academic perspective this will require coding for creating a mobile or desktop application for the user to user. Also, we plan on using a sql database table to store the information, so we will need to use sql to create this. My focus will be integrating the sql database table into our project.

During my experience while in college, I've taken courses that were required for a computer science major. Throughout the years I've learned about different coding languages such as python, c++, matlab, java, and sql. The courses that taught me these languages were Engineering Models 1 (ENED1090) and 2 (ENED 1091), Computer Science 1 (CS1021C), Python Programming (CS2021), and Database Design (CS4092). Other courses taught me the fundamentals for computer science, such as Data Structures (CS2028C) and Discrete Structures (CS2071), which taught me about storing data, and D&A Algorithms (CS4071) and AI: Principles and Applications (CS4033), which taught me about algorithms and what to use to

search through data. Before learning about these I had to have a basic understanding of math by taking Linear Algebra (MATH2076) and Calculus 1 (MATH1061) and 2 (MATH1062). Taking all of these courses have build the foundations of my understanding the computer science world. These courses would also help me during my co-op experience.

During my co-op I worked at Fortech LLC for all 5 co-op semesters. I worked as a junior developer and I was able to learn more things about coding through my senior developers and my supervisor. The main coding languages I would use are javascript, C#, CSS/HTML, and SQL to create small web and desktop applications and perform bug fixes. I was able to implement the fundamentals that I've learned from my computer science courses to create these applications. I was also able to develop my communications skills by communicating progress and issues to my supervisor and asking senior developers for help or advice on projects that I was working on. Through my co-op experience I was able to learn what was expected from me for a person working in the computer science industry.

For my senior design project I am excited about what we are trying to make because I will be able to implement some things that I've learned in my co-op into the project. Our project will be using a sql database to store data and send that information to the user by using a raspberry pi. I plan to use sqlite3 to store a local database into the raspberry pi so it doesn't have to communicate to a database server. For communicating with the user and the database I plan to use c#. I plan to use a library called dapper that can allow me to store sql procedures in the project for the c# code to call and get the data as an object. These are the plans that I have for helping my team in the senior design project.

The expected results of this project will be for our project to be able to use a raspberry pi to receive data from a plant water sensor and send it to the user. We want to create an application, either a desktop application or mobile application, for a user interface. I will know when we are done when we have implemented these components. I will evaluate myself based on being able to implement a local database that is able to send data to the user. I will have done a good job if I can get the local database working on the raspberry pi and I will be done if I can get it to communicate with the user. Overall I think my teams project has an achievable goal and implements a lot of components in computer science.

Kevin Eaton

This semester has taught me that if anyone is as busy as I am right now, taking care of a plant is the last thing on some people's mind. I personally have killed too many plants from forgetting to water them or not giving the proper amount of sunlight. Our senior design group project focuses on developing an Android app, a database, and hardware to monitor water levels and light levels for a potted plant. These water and light levels will be sent to the app through the database for the user to monitor to see if their plants need any more water or sunlight and aid plant owners in their day to day lives. This project will allow us to learn how to integrate

hardware and software into one solution using all the tools we have learned from my four years of school and co-op experiences.

Many of my college experiences will help guide us in this project's development. A lot of the classes I have taken over the past four years and currently are building blocks to my current knowledge base. Some specific classes that will help are:

CS4092 *Database Design/Development*:

The database design and development course was helpful to understand the basics of SQL and the base functions of how databases operate. This will be useful to us in designing our database to interact with the android application and the hardware components.

EECE 3093 *Software Engineering:*

The software engineering course was a good introduction on how to work together in a group and different strategies on how to collaborate on a software project. The tools that we learned about and the work we collaborated on during the class will be helpful in our senior design project.

CS 2021 *Python Programming:*

Along with any other programming course we took, python programming will be instrumental in making this project work. This course set the foundations of how python works and is important for use in our Raspberry Pi, which works nicely with the GPIO pins.

All other classes, such as data structures and algorithms, are also foundational in my understanding of how to write code effectively and efficiently. These do not necessarily contribute directly to this project's development but are fundamentally a part of my understanding of software development.

In my co-op semesters, I learned a lot of technical but also soft skills that are applicable to this project. Specific examples of that are:

Ariel Corporation as a Mobile App Development Co-op:

As a mobile app developer, I learned a lot of technical skills related a lot to developing for Android and Apple. This will benefit this project with working in android to develop an app to communicate with our plant monitoring hardware. Along with technical skills, this co-op was great for soft skills. It was my first development job in a corporate environment and I learned how to collaborate well on software projects with others. This will help in our project with collaboration between my team members.

Booz-Allen Hamilton as a Computer Engineering Co-op:

In my co-op at Booz Allen, I worked a lot with Python and Linux as well as some hardware integration. This will be very helpful to integrate the hardware with the software as we plan on doing. A soft as well as technical skill I gained on this co-op was collaboration through GitHub. I had never worked with others fully virtually before and shared code through GitHub and it poses both a technical and communication challenge to make sure no one erases others progress and that everyone is working on their part.

This project excites me a lot. Doing an IoT project has always been something I have wanted to fully complete and this is a great chance for me to try it out and do some fun work. I have had a raspberry pi for a while now and am just now getting around to use its full potential. I think IoT has plenty of applications for almost everything and learning the challenges and methods of an IoT solution now will be important for my future. The preliminary approach for me when designing a solution is to start big and go small. Taking the big picture and slowly breaking it down into more and more detailed components, like a tree data structure, will eventually get to every small detail of the project. This is my tactic to approach almost every problem and allows for an organized and efficient way to understand the problems and what is needed for the solutions. The expected results are an app that effectively can read data from the Raspberry Pi sensors from the database. The app should also be able to automatically water your plant as well as let you know if more sunlight is needed to be grown effectively. For a stretch goal, adding either photo recognition of a plant to let you know if it has the proper water and sunlight based on the species would be interesting to do. For myself, I will self-evaluate my contributions based on if the hardware to software integration works well. This is because I have made it my main responsibility to make the hardware work with our other components (app, database). If the communication between database, app, and hardware all work reliably and efficiently, I will count my contributions as successful.

Professional Biographies

Christopher Butts

Contact Information

- University email: buttscm@mail.uc.edu
- Phone number: (614)-530-0414

Co-op Experiences

IT@UC | IT Support Technician | Spring 2019, Fall 2019

- Communicated with between 15 - 30 clientele each day to solve PC related issues
- Performed hardware diagnosis and hardware replacement on machines provided by clientele
- Helped maintain computer labs and printers across UC's campus

Martin & Associates | CS Technician | Spring 2021, Summer 2021

- Wrote utilities and enhancements for clientele's Sage100 systems. Enhancements and utilities would consist of changes to backend tables, and changes to the UI
- Communicated with clients to get a list of requirements for their project, and completed the project in a timely manner
- Occasionally worked an IT queue to help clientele with Sage100 related problems

Computing Skills

- Languages: C#, Python, Java, ProvideX
- Operating Systems: Windows, Mac
- Database: SQL, XML
- Miscellaneous: Office Suite, Photoshop

Raymond Gee

Contact Information

- University email: geern@mail.uc.edu
- Phone number: (440)-570-1777

Co-op Experiences

Developer, Fortech LLC, West Chester, Ohio | Spring 2019, Winter 2019, Summer 2020, Spring 2021, Summer 2021

- Worked on developing a web browser and a windows form application for adding, editing, or deleting records in a SQL database
- Designed and edited web browser pages with Javascript, CSS/HTML, or ASP .NET
- Worked on application for automatically sending text messages to a user's phone on completion of a task
- Converted a report builder from using activereports to HTML to avoid issues with users not having licenses
- Performed various bug fixes for existing applications

Computing Skills

- Programming: C++, C#, Python, Javascript
- Web Development: CSS/HTML
- Database Programming: SQL
- Office Applications: Microsoft Office Suite (Word, Excel, PowerPoint)

Kevin Eaton

Contact information

- University email: eatonko@mail.uc.edu
- Phone number: (513)-498-7377

Co-op Work Experience

Computer Engineering Co-op | Booz-Allen Hamilton | August 2020 – August 2021

- Worked independently to develop scripts to automate an open-source tool used for designing microelectronics.
- Developed Python scripts, utilizing machine learning algorithms, to learn from previous program outputs and adjust inputs to converge on the most optimized design parameter.
- Set up and taught the microelectronics team how to use GitHub to efficiently share code internally and across Booz-Allen.
- Collaborated on multiple team projects focused on microelectronics, learning advanced microelectronics concepts quickly to contribute to the team.
- Obtained Secret Level Security Clearance.

Software Development Co-op | Ariel Corporation | May 2019 - May 2020

- Solely responsible for developing a functioning, updated version of the _Ariel 7-2Go_ Mobile Application, utilizing React Native mobile development framework.
- Extracted core functions of the previous application, converting those functions into the new framework.
- Designed a new layout of the new application to increase efficiency and improve user experience.
- Collaborated with sales team members and field users of previous application to gain insight into how to improve the application.

Research Co-op | University of Cincinnati | August 2018 – April 2019

- Worked with Lead Researcher, Nabil Nassif, Assoc. Professor, on Using Computational Intelligence for Building Energy Efficiency.
- Completed thorough literature search to find various types of data-based models that applied to modeling building energy system data.
- Analyzed energy data, developed regression models, created a saving calculation model, and tested modelling techniques using MATLAB.

Summer Intern | Cintas | May – August 2017, May – August 2018

- As part of a multimillion-dollar SAP project initiative, worked directly with the material data steward group to research vendor material information for this Fortune 500 Company.
- Assisted in analyzing the current material database to identify anomalies and present options for remediation.
- Became proficient in both Access and Excel while learning SAP data management tools.

Computing Skills

- Programming: C++, C#, Python
- Web Development: CSS, HTML, Javascript, React Native
- Database Programming: SQL, PHP
- Miscellaneous: Linux, Git

Budget

There have currently not been any expenses. However, it is expected that we spend below $50 on the entire project. Kevin already owns the Raspberry Pi, so the only things we need to acquire are the sensors and the plant. The sensors should be relatively cheap, and the rest of the budget can go towards the plant.

Appendix

Proof of 45 hours

Using the timeline and the effort matrix, If we take out the purchase materials tasks and focus only on the ones where we will be actively working on project components, the lowest effort given by any team member is 30.31% by Raymond. From here we can look at Raymond's most important task: Creating the SQL database. Raymond will have from December 11th, 2021 to February 1st, 2021 to finish this task, giving him a total of 53 days to finish this task. From this one task alone, making the assumption that Raymond spends at least one hour each day on this task up until the final day, Raymond will be spending 53 hours working to create the SQL database. Even if Raymond does not spend all this time working on creating the database, this is only one of the 6 tasks where Raymond is the primary team member working on it. Without even accounting for the other tasks, it is safe to assume Raymond will be putting forth at *least* 45 hours of work on his component of the project.

Following this same methodology for each member, it can be assumed that every member will have at least 45 days where a task is assigned, and working off our 1 hour a day assumption, all of us will spend at least 45 hours working. Besides this though, if Raymond's average effort on the consistent working tasks is 30.31%, then assuming that Kevin and Chris's effort averages are

higher, it can also be assumed we'll be putting forth more effort. Now calculating Chris and Kevin's average effort values, Kevin's average effort is 33.44% and Chris's is 36.25%. Seeing both of these values are higher, I think it can be confirmed that all three of us will be putting forth at least 45 hours of effort into this project.

Github Repository

GitHub