ECE 445

Spring 2023

SENIOR DESIGN LABORATORY

DESIGN DOCUMENT

"Don't Kill My Plant" Habit Tracker

Team 28

Ben Wei (btwei2), DK Ehiribe (ehiribe2), Zade Lobo (zlobo3)

TA: Selva Subramaniam

Professor: Viktor Gruev

TABLE OF CONTENTS

INTRODUCTION

Problem

We are trying to solve a problem that has plagued people for ages: breaking bad habits and adopting good ones. Even though humans may want to change these habits, they usually lack the willpower in order to do so. Common solutions on the market to help people change habits include smartphone tracking apps and physical devices that track physical habits. These solutions are great for tracking, but most of them can be circumvented easily and don't hold people accountable for their actions. In addition, any positive reinforcement methods that they provide are minor and are not effective enough.

Behavioral therapy has been promising in the field of medicine recently, and it is usually used to address traits in a person that might be self-detrimental or harmful for a person to have. Being able to understand what a person values and adapting their thought processes to fit what needs to change can benefit a person in the long term. Bringing behavioral therapy to fixing habits has been tried in the past, but not to as great of an extent as needed.

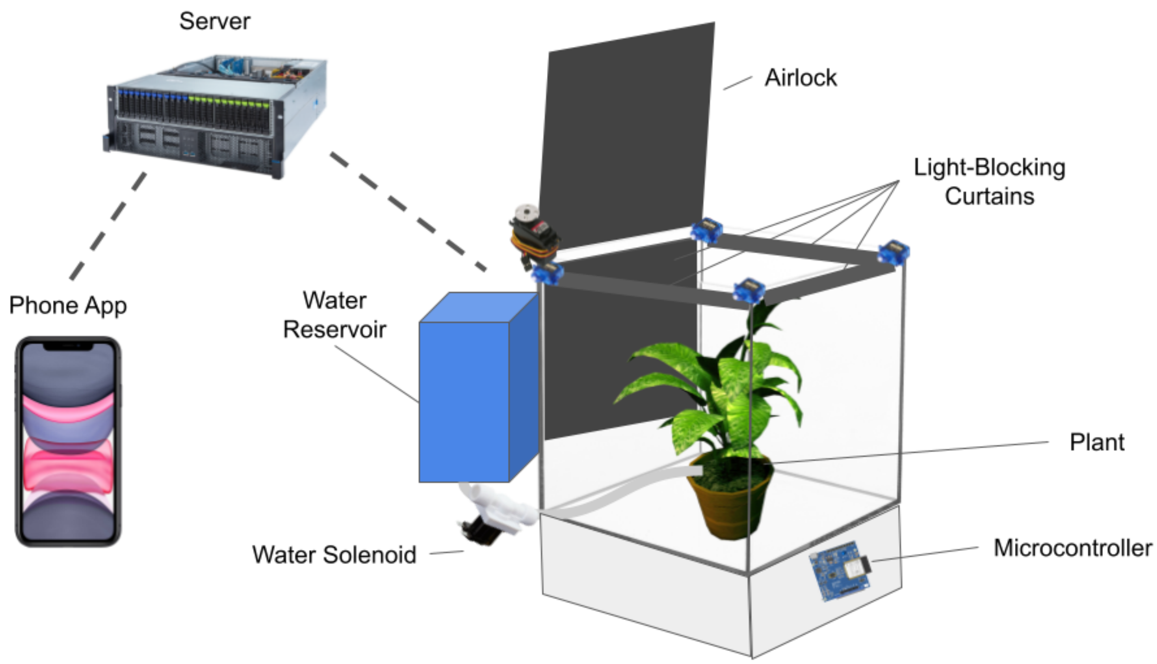
Solution

We want to change this by bringing in emotional attachment to tangible consequences to encourage people to keep up with their habits. While positive reinforcement may not be as effective, negative punishment has also shown promising results. "Don't Kill My Plant" is an application interface that will keep track of your habits through unforgeable data and will make the life of your plant dependent on it through hardware. The solution we are providing is innovative by causing people to emotionally attach themselves to keeping up with good habits as well as keeping a physical and visual reminder.

The concept around the project was to liken personal growth to the growth of a plant. Plants grow slowly, but over time they grow to be much bigger and stronger than when they started. The plant's steady journey is meant to mirror the journey of people on the path to starting better habits and breaking harmful ones. The ups and downs on the journey of habit change are slow, just like the change that our plant enclosure will provide. Because of this, we believe that a house plant is the best symbol for personal change as well as the best means to cause emotional attachment to your goals.

We aim to create a network-enabled plant enclosure that contains all the necessary mechanisms to facilitate or impede plant growth. This would include light-blocking curtains, an airlock, and an irrigation system. Along with the companion phone app, users can track their habits with real data from their phone such as screen time or location-based data. If habits are followed, the enclosure will stay open, curtains will roll up, and the plant will be watered. As soon as good habits are broken, the airlock will seal tight, the watering will stop, and the curtains will roll down.

Visual Aid

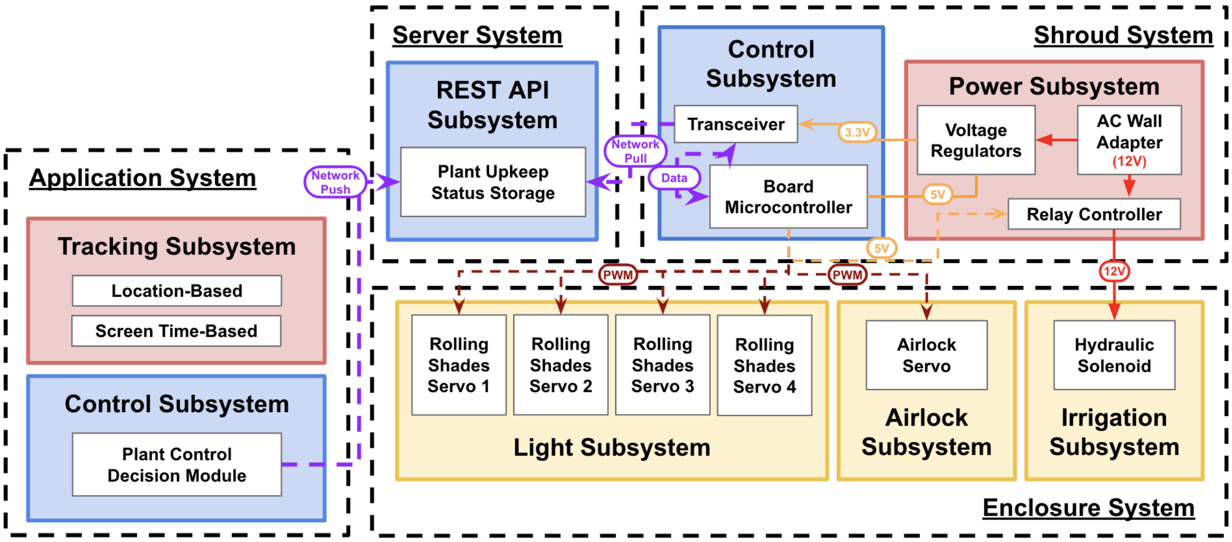


High-Level Requirements

- The application interface is able to facilitate habit tracking for the user and send this information to the physical device
- The created device system is capable of keeping a potted plant alive and killing a potted plant according to the data passed to it.
- The design has a modular design, allowing for more plants-enclosure addons to track more habits.

DESIGN

Block Diagram



Our design receives power from the power subsystem and begins with our application interface subsystem. Here, habits are tracked and the information to determine the fate of the plant is sent to a server and communicated to the plant through the micro controller subsystem. The plant enclosure subsystem then uses this information provided to choose between keeping the plant alive or harming it. The irrigation subsystem is also controlled by the microcontroller and routinely waters the plant when told to keep it alive.

Subsystem Overview

Power Subsystem

This module should interface between power from the wall input and each of the subsystems. This means converting the 12V input to both 5V and 3.3V for other systems.

Requirements	Verification
-Must convert 12VDC to 5VDC	-Check with a multimeter that 12VDC is received from the wall adapter -Check with a multimeter that 5VDC +/-20% is being outputted -Check with a multimeter that at least 0.8mA is being outputted
-Must convert 12VDC to 3VDC	-Check with a multimeter that 12VDC is received from the wall adapter -Check with a multimeter that 3.3VDC +/-10% is being outputted -Check with a multimeter that at least 0.8mA is being outputted

Application Interface Subsystem

The application interface is a phone application that will offer multiple ways to track habit forming, including location, screentime, and message and call tracking. This allows the application to pick up on habits such as going to the gym, avoiding a coffee shop, spending too much time on social media, or messaging your family.

Requirements	Verification
-The application includes basic user configurable habit tracking that leverages phone features such as location info	-Check that the user can input a habit to be tracked -Observe that there is a clear display of whether habits are kept or not
-The application should be able to post binary signals that correspond to each plant enclosure: 0 for habit kept, 1 for a habit broken	-Use a test program to force signals of all 0 and then all 1. Verify that the server signals are updated accordingly. -Use the actual program to test that the habit tracking translates into updated server signals.

Plant Enclosure Subsystem

The plant enclosure is a box with an airtight lid in order to create an isolated environment for a plant. The box itself will contain transparent walls with a method for blocking light out (either electronic tint or rolling window shades. The box will have an airtight lid that can be electronically opened and closed by the microcontroller. This lid and walls are implemented with servos.

Requirements	**Verification**
-The shades can fully roll and unroll through the use of servos.	-First, check that the curtains are mechanically able to be deployed (no snags or other issues) -Next, verify that 5V+/-20% are being received as power -Next, verify that the control signals are correct through the use of an oscilloscope. This involves using a test program to generate component specified PWM signals. -Finally, verify that the system responds appropriately to the test signals. Observe that the servos both fully roll and unroll the shades.
-The airtight lid can fully close and open through the use of a servo	-First, check that the airlock is mechanically able to function -Next, verify that 5V+/-20% are being received as power -Next verify the correctness of the control signal. Use an oscilloscope with a test program, and check that the component specified PWM signals are being generated. -Finally, observe that the airtight lid can be fully closed and opened through the use of these control signals

Microcontroller Subsystem

The microcontroller system will use an ATmega328-P to control the other subsystems. A ESP8266 chip will also be used to add wireless capabilities to the system. This system will regulate the functions of the other subsystems. For each enclosure, the microcontroller will fetch a binary signal from a server and manage systems accordingly. This means, for example, upon receiving a ‘0’ signal for enclosure 0, all subsystems should maintain the plant: regularly water, open curtains, and open airlock. The opposite is true for a signal of ‘1.’

Requirements	**Verification**
-Binary signals are successfully fetched from the server	-Use the application to toggle each enclosure’s binary signals -Observe that the curtain signals activate accordingly with an oscilloscope (which verifies that wireless signals are received).
-Servo subsystems are managed through control signals, opening blinds or closing them and opening the airlock or closing it based on the corresponding binary signal	-First, use a test program to internally force a signal transition from 0 (don’t kill) to 1 (kill) and vice versa (say, transition every given interval of time), this should not depend on testing the wireless capabilities. -Use an oscilloscope to verify the servo specific PWM signals. Check that the width is correct for 0->1 transitions and 1->0 transitions.
-Irrigation subsystems are managed through control signals, watering regularly or withholding water based on the corresponding binary signal	-Test both binary enclosure signals individually via a test program (which forces a binary signal) For a signal of 1 (kill): -Use a multimeter to check that the irrigation control output is never raised to logical 1 For a signal of 0 (don’t kill): -Use a multimeter to check that the irrigation control output is regularly raised to logical 1 for a set amount of time *For example, every hour, the signal is raised

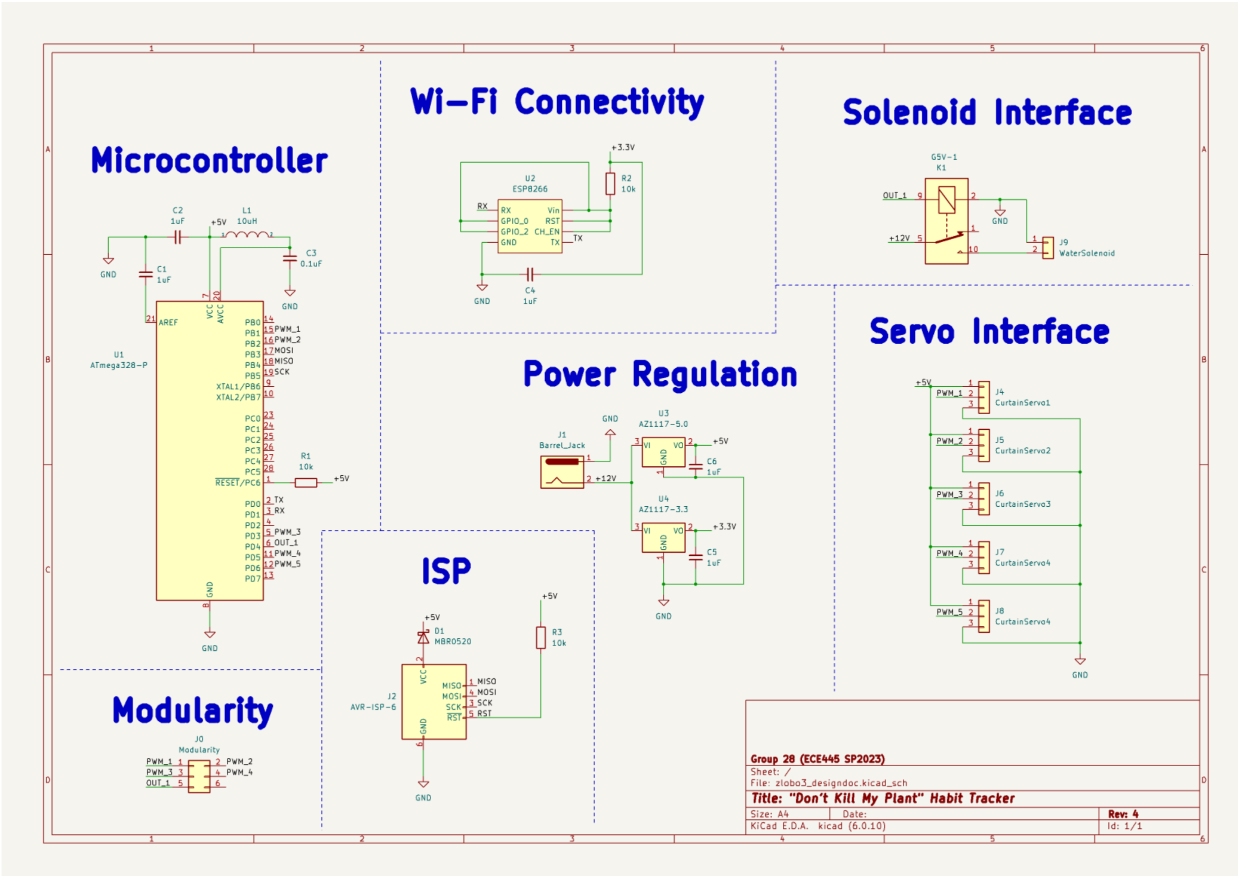
	for 30 seconds (or a shorter interval for testing)

Irrigation Subsystem

The irrigation subsystem will be controlled by the microcontroller in order to routinely water the plant through the included water solenoid. The reservoir outside the plant enclosure will allow the user to input water for irrigation, but the actual water delivery will be controlled through hydraulic tubing piping it inside the system.

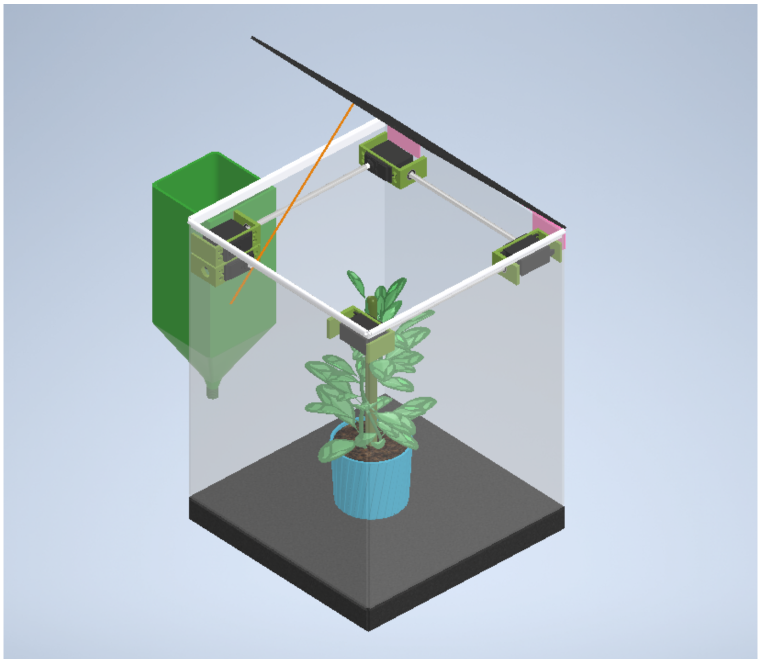
Requirements	Verification
-This subsystem must water the plant when it receives a logical 1 signal and not otherwise	-First, check that the 12V input is functioning with a multimeter -Next, verify the input signals are correct using a multimeter and a test program. The logical high signal should be 5V +/- 10%. The logical low signal should be 0.5V +/- 100%. -Finally, verify that 12V are passed to the solenoid only upon receiving a logical high signal. Again, use a multimeter to verify that 12V are being passed through.

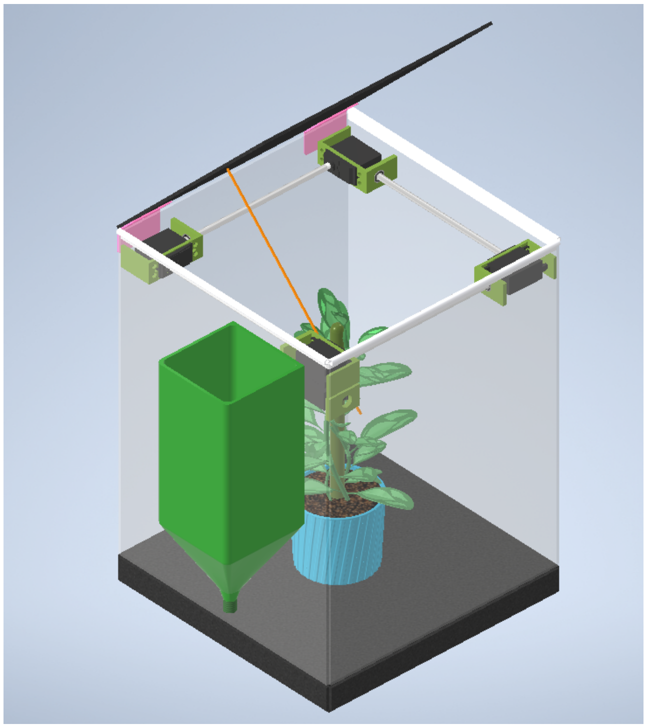
Circuit Schematic

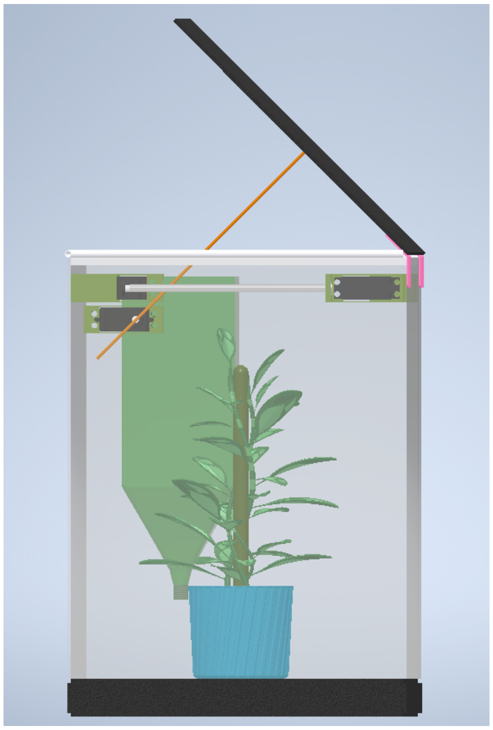


3D CAD Model

An interactive 3D model can be found here.





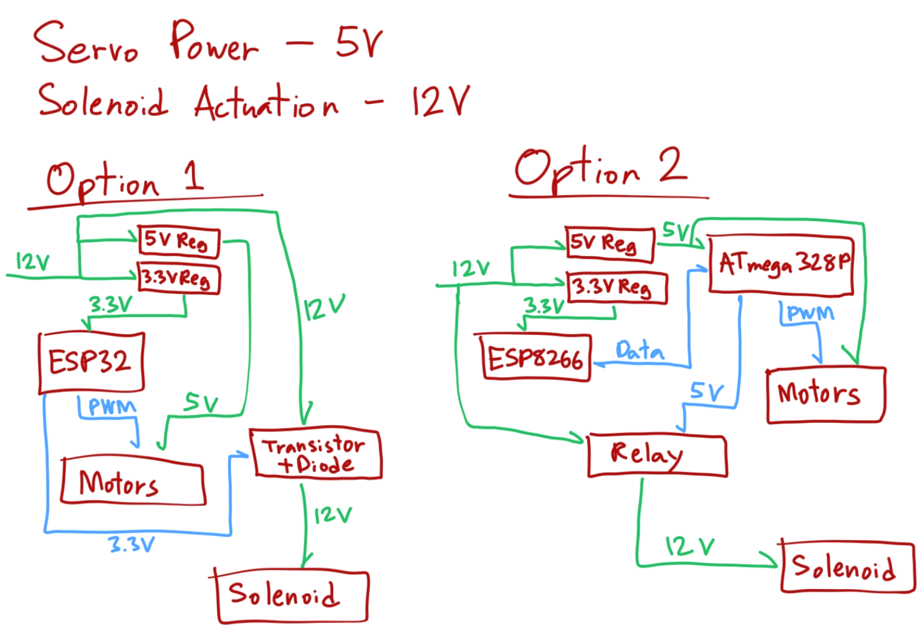


Tolerance Analysis

One of the most difficult sections of the project will be implementing the fluid solenoid. Solenoids inherently require larger voltage sources due to them containing strong inductors for operation. The best solenoid for fluids that we could find operated at 12 volts, which was much larger than the 3.3 volts that we had initially decided on with operation of an ESP32.

An ESP32 chip operates at a maximum of 3.3 volts, causing there to be multiple problems for our use case. The first major problem occurs with the operation of servo motors, which require three connections: 5 volts, a PWM signal, and ground. Our initial thought was to include another regulator for components that needed 5 volts. However, this doesn't fix our second problem, which was the operation of the solenoid. The inductor required a 12 volt digital signal for ideal operation, which the ESP32 could definitely not provide. The alternative solution was to include a transistor with a diode to get 12 volts directly from the power supply, but operating a solenoid with a transistor is not an ideal solution due to the high current and effects of the transistor. We decided to use a relay to operate the solenoid, but most hobbyist relays only work with 5 volt signals and above.

This prompted our switch to using an ATmega328-P chipset, commonly found on boards like the Arduino Uno and Arduino Nano. The chipset can be powered with as low as 1.8 volts, but can use up to 5 volts for its digital logic. With the use of this microcontroller, we downgraded from the ESP32 to the ESP8266 for wireless communication because we only needed the transceiver. This not only allows us to power the servo motors much more easily, but it also allows the inclusion of a relay for use with the solenoid.



COST & SCHEDULE

Cost Analysis

Itm	Qty Unit	Qty	Qty/Pck	Amt/Pck	Pcks Prchsd	Qty Rcvd	Amt
Ball Bearings	Number	4	10	$7.99	1	10	$7.99
Black Acrylic	Number	3	1	$12.60	3	3	$37.80
Vinyl Gasket	Feet	4	17	$6.93	1	17	$6.93
Steel Rod	Inches	42	48	$6.38	1	48	$6.38
Clear Polycarbonate	Number	4	1	$10.24	4	4	$40.96
Silicone Sealant	Number	2	2	$9.56	1	2	$9.56
Blackout Fabric	Yards	1	1	$7.99	1	1	$7.99
ATmega328-P	Chip	1	1	$3.25	1	1	$3.25
MBR0520	Component	1	1	$0.02	1	1	$0.02
Barrel Jack	Component	1	1	$1.09	1	1	$1.09
AZ1117-5.0	Component	1	1	$0.40	1	1	$0.40
AZ1117-3.3	Component	1	1	$0.40	1	1	$0.40
G5V-1	Component	1	1	$2.31	1	1	$2.31
ESP8266	Component	1	1	$7.50	1	1	$7.50
Pin Headers	Component	20	1	$0.35	20	20	$7.00
10K Ohm Resistors	Component	3	1	$0.10	3	3	$0.30
1 uF Capacitors	Component	5	1	$0.10	5	5	$0.50
0.1 uF Capacitors	Component	1	1	$0.36	1	1	$0.36
900-00008	Component	5	1	$18.95	5	5	$94.76
AC Power Adapter	Adapter	1	1	$10.88	1	1	$10.88
997 Water Solenoid	Component	1	1	$6.95	1	1	$6.95

In addition to the cost analysis above, we are also including a time-cost for employment of the people working on this project. We assume that 3 students are working on this project at a pay of $41 per hour. With a total of 40 hours working on the project and an overhead of 2.5 times the hourly rate, we arrive at a labor cost of **$12,300**.

Subtotal	$253.33
Expected Tax	$22.80
Expected Total	$276.13
Including Labor	$12,576.13

Schedule

Week	Task	Person
2/20-2/26	Circuit Schematic Design	Zade
	Team Contract	All
	Design Document	All
2/27-3/05	Finalize PCB Design	All
	Order PCB	
	Order COTS Parts	
3/06-3/12	Develop Mobile App	Ben
	Collect Parts	Dike
	Fabricate Enclosure	Zade
3/13-3/19	Spring Break	
3/20-3/26	Solder PCB	Dike
	Develop Mobile App	Ben
	Fabricate Enclosure	Zade
3/27-4/02	Solder PCB	Dike
	Fabricate Enclosure	Zade
	Finish Mobile App	Ben

4/03-4/09	Finalize PCB	Dike
	Wiring of Enclosure	Zade
	Write Code for Board	Ben
4/10-4/16	Integration and Debugging	All
4/17-4/23	Mock Demonstration	All
	Presentation Rough Draft	
4/24-4/30	Demonstration	All
	Presentation Final Draft	
5/01-5/04	Final Presentation	All
	Final Paper	

ETHICS & SAFETY

A potential safety concern that may arise in this project would be the exposure of certain parts of our system to water delivered through the irrigation subsystem. We can address this issue by ensuring that the plant enclosure system is properly sealed and can be properly drained to avoid leakage into other components. We plan to be mindful of the IEEE Code of Ethics 7.8.I.5 as we will be receiving continuous feedback through the development of this project and we will use this feedback to improve our project and continue producing honest data. Relating to the IEEE Code of Ethics 7.8.I.1, we also plan to be mindful of the safety of the public and the environment by ensuring that nothing other than a plant is placed into the airtight plant enclosure system because this system can cause harm to the object it encloses.

Citations

"IEEE code of ethics," Institute of Electrical and Electronics Engineers (IEEE), Jun-2020. [Online]. https://www.ieee.org/about/corporate/governance/p7-8.html. [visited on 09-Feb-2023].

"Atmega328 Pinout," *Atmega328 pinout*. [Online]. https://www.learningaboutelectronics.com/Articles/Atmega328-pinout.php. [visited on 21-Feb-2023].

B. Tymrak, "Programming an AVR ATmega328P with an Arduino," *Programming an AVR atmega328p with an Arduino*. [Online]. https://www.brennantymrak.com/articles/programming-avr-with-arduino. [visited on 22-Feb-2023].

"Transistors, relays, and controlling high-current loads," *ITP Physical Computing*. [Online]. https://itp.nyu.edu/physcomp/lessons/electronics/transistors-relays-and-controlling-high-current-loads/. [visited on 20-Feb-2023].