**SOFTWARE  
DEVELOPMENT PROPOSAL**

### 

### 

### 

### 

### 

### 

### **PROJECT NAME**

AutoFarmOS

### **PROPOSAL DATE**

October 28, 2019

### **DESIGN TEAM (TEAM #7)**

Matthew Cherry

Kyle Curry

Kristi Daigh

Ethan Lefert

Zach Freund

# 1. Project Overview

###### **Synopsis**

AutoFarmOS is the software used to control the sensors and actuators for the Terrafarm farming module.

###### **Description**

This project is being undertaken to satisfy the embedded software needs of the farming module for the start-up company, Terrafarm. The start-up is currently working to establish a minimal viable product to prove the viability of their idea and our team hopes to support this goal by developing an operating system and drivers for the module. The Autofarm product aims to solve many agricultural and environmental issues. According to Terrafarm,

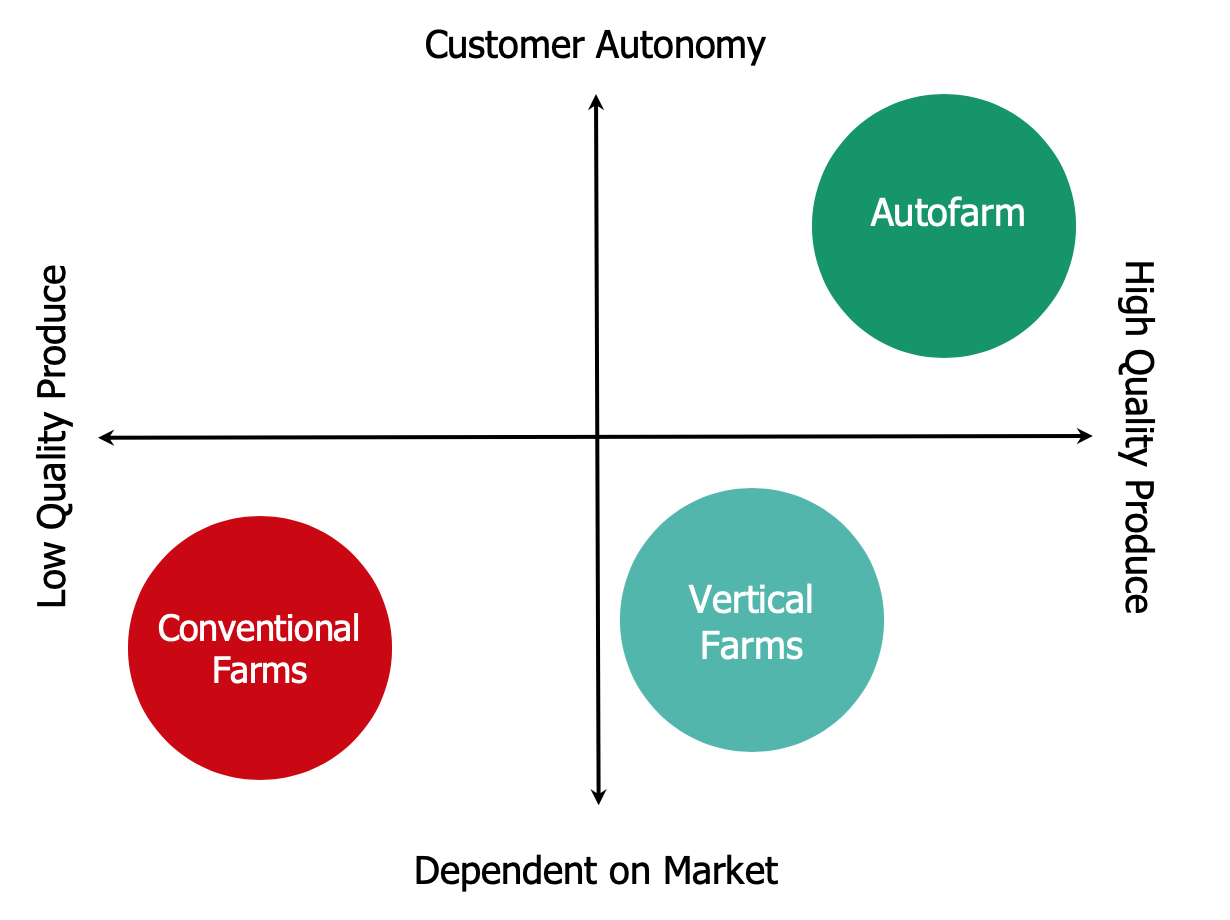
*“Autofarm is intended to completely disrupt the food supply chain by introducing 21st century information technology to agricultural production. This will eliminate substantial food waste, reduce environmental degradation, improve produce freshness and nutritional content, increase food security and accessibility, slow the process of climate change, and make cities more sustainable and self-sufficient.”*

By working with Autofarm, we hope to help push the world towards greener forms of agriculture. Additionally, we as a team are excited about the opportunity to explore new development territory by working with embedded systems and by undertaking a large, multi-team project. The end result for our project is a minimum viable embedded software system that can be combined with other software components to form a comprehensive software suite that can be extended for use in Autofarm commercial units.

# 2. Project Design

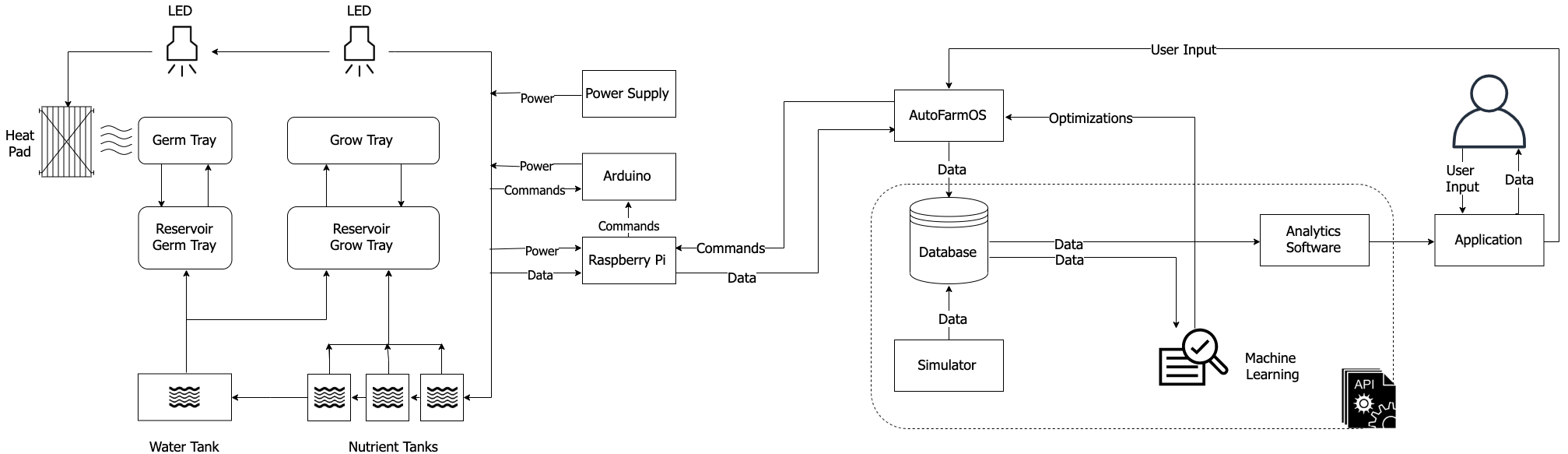
###### **Product Overview**

The product, as designed by the start-up company Terrafarm, has two main components: a farming module and a software suite. The farming module is a small-scale, refrigerator-sized vertical farm that will, by the product description, “autonomously grow fresh, nutritious produce for food businesses such as grocery stores and restaurants.” The software suite has three primary functions: handle data from the farming module, control growing conditions in the farming module, and provide an interface for users to interact with the module. Together, the hardware and software components form a system that yields high quality produce while also giving produce-dependent businesses more control over how their produce is grown and handled.



**Figure 1: Autofarm Market Analysis Graph**

The product design team is composed of the start-up company itself, which serves as the product owner, the mechanical engineering team, which is responsible for designing the hardware unit, and three software teams, which are responsible for user interface, data analytics, and embedded systems, respectively. The user interface team is working to develop a website (that may later be extended to a mobile application) to allow users to interact with the farming module by viewing data and sending commands. The data analytics team is designing a machine learning program that will use sensor data to ultimately determine settings and inputs for better plant growth. Our team, the embedded systems team, is developing the software that will read sensor data and control actuators, by sending information to the server and receiving commands from the server, respectively. All of the software teams will work together to establish proper interfacing methods for the separate software components.



**Figure 2: Product Overview Scheme**

###### **Hardware Design**

The hardware involved in our project will consist of two systems. Both parts will be using a Raspberry Pi 4 as the backbone of the embedded system. Part one will include the sensors connected to arduinos which then report to the Raspberry Pi 4. Part two will control the pumps, pressure valves, and any other mechanical systems that need a controller. We chose to separate the functionality of these two systems for two reasons. The first reason being the amount of ports available to us. Having two Pi’s will allow us to organize the internal wiring ergonomically. Secondly, we wanted to have a failsafe for if one system goes down. Being as our project is based off of the growth of plants, it may be awhile before anyone would notice that something has gone awry. If one system were to go down for any reason we would be able to use the second Pi as a failsafe to let the server know this unit is malfunctioning.

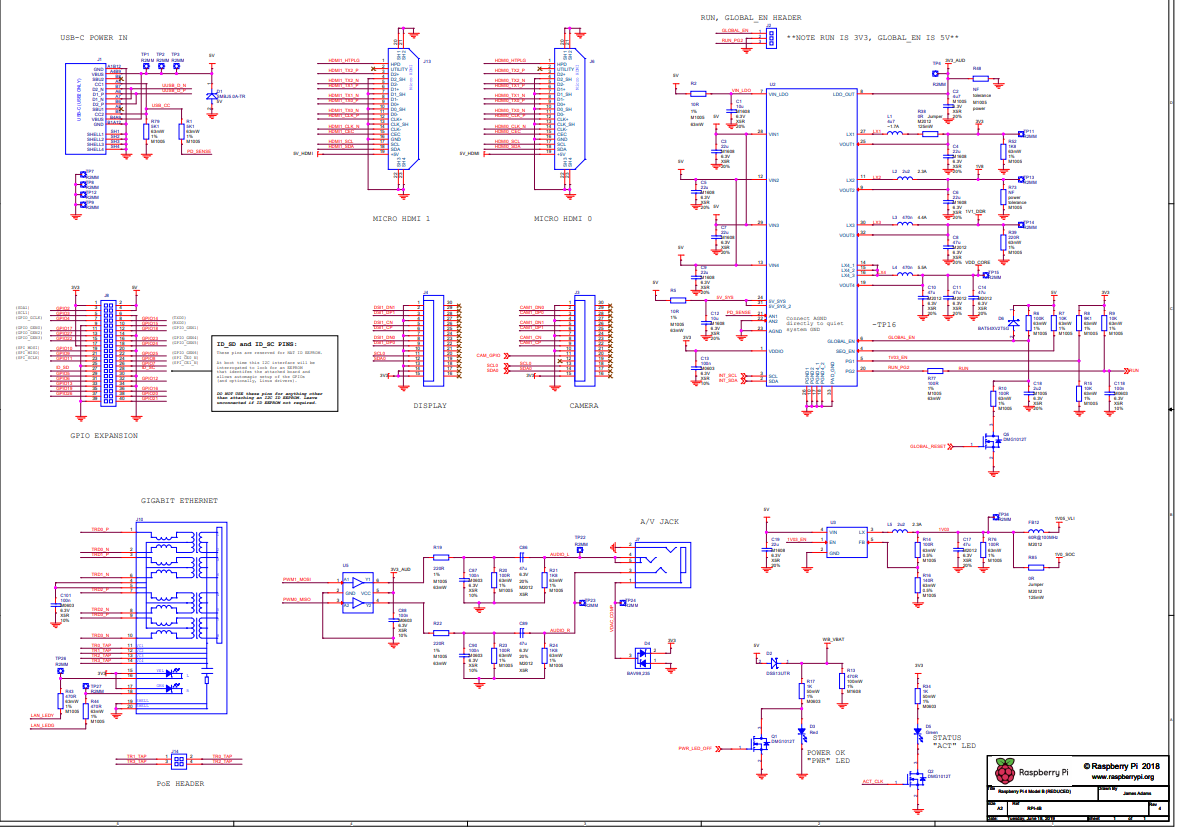
The first system will be configured as a star topology connecting all of the sensors we need to a Raspberry Pi 4. The sensors will first communicate through an arduino and the arduinos will be wired over USB or I2C (depending on the sensor) to the Raspberry Pi. The Pi will be the main computer for our project. It will handle many of the information passing tasks: collection, organization, and uploading to the server. The model of Raspberry Pi we have chosen included 4GB of memory allowing us ample computing power to do the collection of data while using it to communicate to the server or user.

The types of sensors we will be using will measure PPM (Parts Per Million), pH, temperature, humidity, and light wavelength. To measure the temperature and humidity we will be using a hygrometer that is water resistant to withstand the residual condensation from the misters that water the plants. It will be connected to an arduino in a waterproof case using wiring safe for use in wet environments. To measure PPM and pH we will have two separate probes that will be used in the water tanks to evaluate whether or not adjustments need to be made to the water. Lastly to measure the wavelength of light that our LED’s are emitting we use a PAR (Photosynthetically Active Radiation) which will use an arduino uno to report back to the sensor Pi.

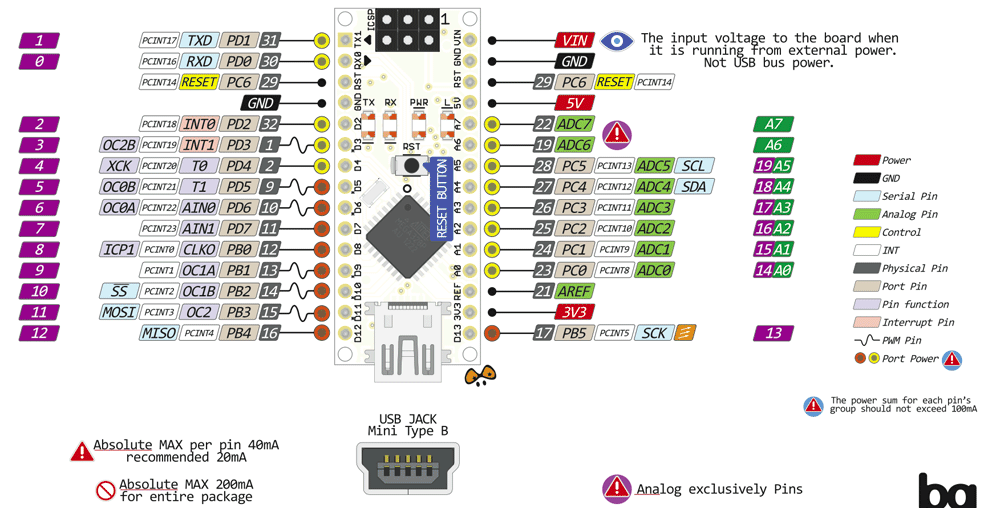
The second system we will be implementing is a control system for the tanks, misters, thermal pad, and LEDs. The LEDs we will be using are called Cob LEDs which will allow us to configure them in use for growing. A thermal pad will control the heating of the germination tray ( a special part of the hydroponics unit that will be tailored for seed germination). Lastly, we will control the rate at which the water system disperses water.

Our task will be to implement an embedded system to react to the readings in our sensors and the parameters given to us by the server. We will implement this functionality using arduinos nanos/relays to adjust the heat of the thermal pad, the wavelength given off by the LED bulbs, the frequency of the misters and to alert to a low tank water level. The requirements of the control system will change as we see progress from the other teams. Our goal is to make a robust design that will allow for changes later on.

In conclusion, our sensor system will gather data relating to the wellbeing of the plants and their environment. We will then adjust our control systems to match the predefined configuration given to us from the server and the current readings from our sensor network. We will then send our data to the AutoFarmOS server.



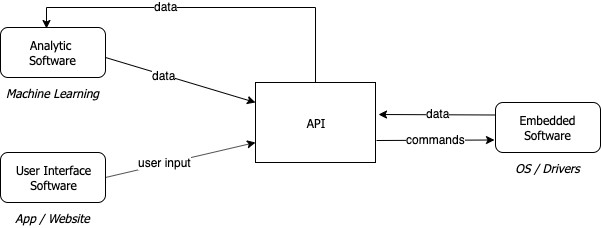
**Figure 3: Raspberry Pi Diagram**

****

**Figure 3.5: Arduino Nano Diagram**

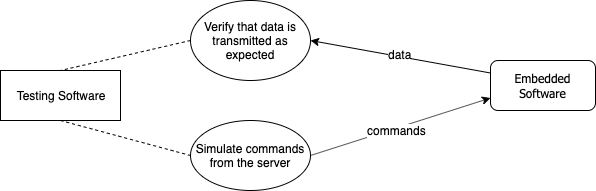
###### **Embedded Software**

The software for this project will be fairly simple, consisting primarily of a base operating system for the module, drivers that will connect hardware and software, and API wrappers used to communicate with external software components. The base operating system will be open-source, selected by our team based on hardware compatibility. The drivers, which will either be provided by the sensor manufacturers or written by our team, will have two main functions: to get data from the sensors, such as information about growth conditions, and to send commands to the actuators, which will control the LEDs and the watering of plants. The API wrappers will allow the other software components, potentially written in other languages, to interact with the sensors and actuators of the module.

****

**Figure 4: Basic Software Interface Map**

Testing software will also be a significant part of our development. Due to the fact that the development of the user interface software, analytic software, and embedded software are all being done separately, we will have to develop tests that simulate the receiving of data on the server and the sending of commands from the server. This is necessary to ensure that the embedded software is as functional as possible when it is brought together with the other software components of the overall product.



**Figure 5: Basic Testing Software Map**

One of the main software constraints for this project is deciding how the hardware system will communicate with the server over the internet. We will have to decide what the unit will do if it loses connection to the internet. We will also have to decide what it will do if any of the tanks run out of water or nutrients. Another constraint is our team’s limited knowledge of embedded software, which will be overcome with significant amounts of research along with assistance from a faculty mentor. An overall project constraint that will factor into software development is the commercial future of the product. Planning and research is critical since the software developed could potentially be sold commercially in the future. In addition, thorough documentation is critical if the Terrafarm team plans to take over the maintenance and support of the software after the term of this project is completed.

###### **Integration**

**The Autofarm OS we are developing is the heart of the larger project in regards to functionality. Integrating our operating system symbiotically with each branch of the larger project must be considered throughout development. The unilateral functionality provided by our operating system will allow access to sensor data, enactment of actuator changes and allow threshold set points to be instantiated. The “language” of communication is dictated at the lowest-level by our defined API function calls and/or associated API wrapper function calls. Collaboration with all groups is a key part of the projects success.**

The connection between the sensors and actuators and the module’s operating system is the drivers. With regards to sensors, the drivers are responsible for reading in the data, analyzing it, formatting it, and sending it to the server. For the actuators, the drivers are responsible for receiving signals from the server, interpreting those signals, and sending them to the actuators. As mentioned above, the drivers will either be provided by the manufacturer of the sensors or be developed by our team and this will likely vary from sensor to sensor. The most important factor here is the compatibility of the drivers with the operating system. The provided drivers or the driver development process for each hardware component (i.e. sensor or actuator) will be considered heavily during part selection as an optimization of hardware and software codesign. Optimal part selection will allow for the fastest possible development cycle and, because we are working with a start-up company, speed is critical.

# 

# 3. Project Logistics

# **Milestones**

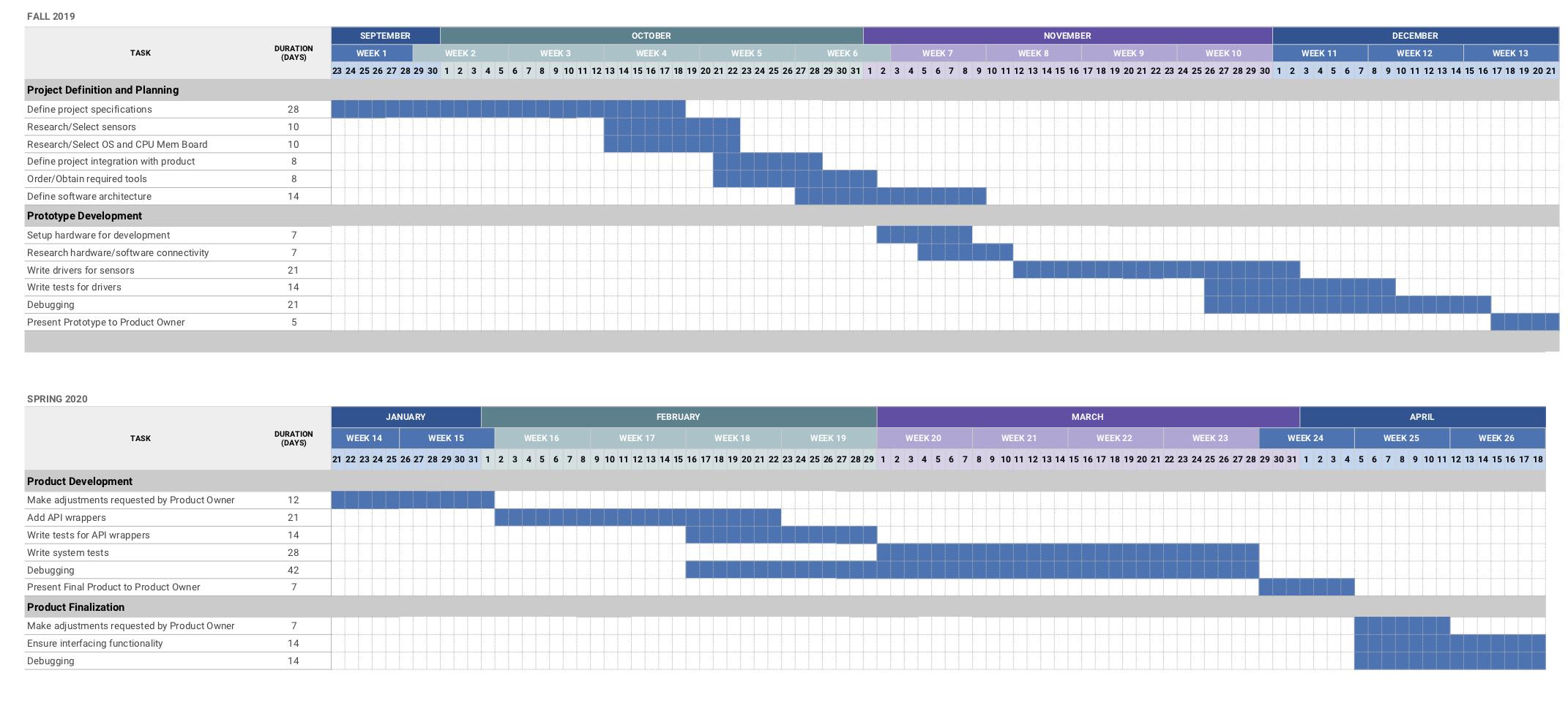
|  |  |  |
| --- | --- | --- |
| **Milestone** | **Tasks** | **Date** |
| **1 - Project Definition & Planning** | | |
| 1.1 | Project specifications defined | 10/18/2019 |
| 1.2 | Hardware specifications defined | 10/22/2019 |
| 1.3 | Integration with overall product defined | 10/28/2019 |
| 1.4 | Required tools obtained | 11/01/2019 |
| 1.5 | Software architecture map established | 11/08/2019 |
| **2 - Development** | | |
| 2.1 | Hardware setups completed for development purposes (one for each team member) | 11/09/2019 |
| 2.2 | Basic HW/SW connectivity established | 11/11/2019 |
| 2.3 | Drivers for sensors completed | 12/01/2019 |
| 2.4 | Basic OS prototype completed | 12/20/2019 |
| 2.5 | API wrappers completed | 02/22/2019 |
| **3 - Testing / Diagnostics** | | |
| 3.1 | Tests for drivers completed | 12/09/2019 |
| 3.2 | Tests for wrappers completed | 02/29/2019 |
| 3.3 | System tests completed | 03/28/2019 |
| **4 - Finalization** | | |
| 4.1 | Alpha Prototype completed; Presented to Product Owner | 04/01/2019 |
| 4.2 | Interfacing with external software components confirmed | 04/12/2019 |
| 4.3 | Modified Alpha Prototype completed | 04/18/2019 |

# 

# **Work Plan**

* Matt Cherry
  + Chief Responsibility: OS Selection
  + Additional Responsibilities: Sensor Selection, Driver Development
  + Sensor Assignment: LED/PAR
* Ethan Lefert
  + Chief Responsibility: Sensor Selection
  + Additional Responsibilities: Driver Development
  + Sensor Assignment: Hygrometer
* Zach Freund
  + Chief Responsibility: Driver Development
  + Additional Responsibilities: Testing
  + Sensor Assignment: PPM
* Kyle Curry
  + Chief Responsibility: Testing / Diagnostics
  + Additional Responsibilities: Driver Development
  + Sensor Assignment: pH
* Kristi Daigh
  + Chief Responsibility: Product Management
  + Additional Responsibilities: Driver Development, Testing
  + Sensor Assignment: Tank Gauge

# **Schedule**

Gantt chart is shown below and also [linked](https://docs.google.com/spreadsheets/d/1Qspj9KhGHmnRaOg2Xw-vIbCidRiLu0_JYu79MSWwa7c/edit?usp=sharing) for better readability.[](https://docs.google.com/spreadsheets/d/1Qspj9KhGHmnRaOg2Xw-vIbCidRiLu0_JYu79MSWwa7c/edit?usp=sharing)

# **Budget**

|  |  |  |  |
| --- | --- | --- | --- |
| **Resource** | **Vendor** | **Date Needed** | **Estimated Cost** |
| Raspberry Pi 4 Model B Kit (x5) | CanaKit(Amazon) | 11/1/2019 | $500.00 |
| SHT20 I2C Temperature & Humidity Sensor (Waterproof Probe) | DFRobot | 11/1/2019 | $22.50 |
| Apogee FullSQ-500-SS: Full-Spectrum Quantum Sensor | Apogee | 11/1/2019 | $378.00 |
| Conductivity Probe K 1.0 | Atlas Scientific | 11/1/2019 | $139.00 |
| Gravity: Analog pH Sensor / Meter Pro Kit For Arduino | DFRobot | 11/1/2019 | $56.90 |
| Gravity: Non-contact Digital Water / Liquid Level Sensor For Arduino | DFRobot | 11/1/2019 | $10.00 |
| Gravity: Water Flow Sensor (1/2") For Arduino | DFRobot | 11/1/2019 | $10.00 |
| DFRduino UNO R3 - Arduino Compatible | DFRobot | 11/1/2019 | $15.00 |
| Arduino Uno (x4) | Arduino | 11/1/2019 | $60.00 |
| AmazonBasics USB 2.0 Printer Cable (x4) | Amazon | 11/1/2019 | $20.00 |

# 

# **Ethical and Intellectual Property Issues**

Ethical Issues:

One ethical issue we may face in regards to our project is the security of controlling the farming module. Since the user will be able to control the module through an app or website, we have to authenticate that only permitted users have control. If someone else were to gain control of the machine, they could destroy the crop currently being grown resulting in a loss of time and money for the owner.

An ethical issue that our project is addressing is over farming. There are also issues of topsoil erosion and overuse of pesticides and fertilizers leading to soil pollution. The concern is that eventually we won’t be able to keep up with demand due to damaging the growing conditions. The hope is that the TerraFarm farming module can help reduce the environmental effects of current farming methods.

Intellectual Property Issues:

One issue we have is related to how to handle our portion of the project once we have finished our work. As we are developing the operating system for a start-up, the company will most likely want to take ownership of the intellectual property, assuming it functions properly and meets their expectations. If this proves to be the case, our team will have the option of selling the software to TerraFarm which would involve a purchase of the intellectual property.

Another issue we may face is the use of certain drivers for controlling the sensors and actuators. We have to be sure that we are allowed to use the drivers that we choose. We don’t want to use any drivers that aren’t open source. When we pick out what we want to use, we also need to be sure that we use the drivers in a means that are permitted by the original creator. As we are purchasing various parts for the farming module, these will include drivers that will help us avoid these problems.

# **Change Log**

As a whole, our project is going to be quite different than the initial description. After meeting with the product owner, our team’s portion of the project changed to developing the operating system for the Terrafarm farming module. The first change we made was the name of our project from TerraFarm App to AutoFarmOS. In addition to a name change, our project synopsis and description had to be changed to better reflect the fact that we are no longer working on an application to control the farming module. The project milestones and work plan both had to be updated as the initial milestones and work division were related to progress for application development. Finally, our budget was changed to now include the costs of the various parts and sensors needed to complete our project.