Team 16
McKenna Groves, Adam Khan, Quinn Meier, Evan Trout, Mark VanLandingham

Project Name
Autofarm Website and Progressive Web Application

Project Synopsis
Progressive web application that connects customers to their Terrafarm modules for real-time monitoring and control.

Project Description
Autofarm is part of the pitched project Terrafarm, which describes its goals as such:

“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. Imagine an application that fulfills this purpose while enabling stores to easily grow their own food, saving them money, attracting customers, and eliminating supply issues. This application, Autofarm, will operate tiny vertical farms that will autonomously grow fresh, nutritious produce for food businesses such as grocery stores and restaurants.”

The Autofarm applications will provide Terrafarm customers with a powerful and easy-to-use interface for monitoring and controlling their Terrafarm units. These applications will provide data, progress, and alerts regarding the user’s Terrafarm modules, and connect to the cloud to share data with other modules.

Due to the size and ambition of Autofarm, our group will be working in cooperation with other groups who will be handling the creation of a database, operating system, and machine learning platform for the initial iteration of the Autofarm.

Project Milestones
<u>First Semester</u>
1. Plan design of application with Terrafarm directors
2. Choose technology stack

<u>Second Semester</u>
1. Complete minimum viable product of progressive web application (March)
2. Integrate with other Autofarm projects (April)
3. Complete final builds (May)

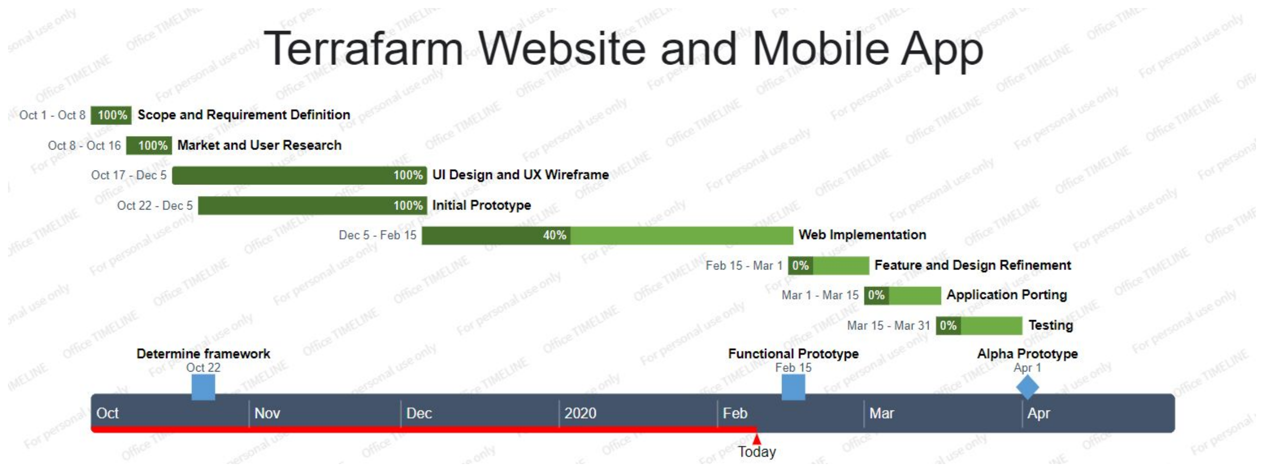
Project Budget

The only cost for our project will be whatever cost is associated with hosting the site, which won't be determined until later in the project.

Work Plan

Our goal is to evenly divide work as much as possible so that all team members get exposure to the technologies being used to develop this web application. All of our team members have some degree of familiarity with web development, but Mark VanLandingham is taking the lead on development due to his experience with the specific technology stack we are using.

Gantt Chart



Preliminary Project Design

Our team is creating the application for the end user to view and manage his or her Terrafarm(s). We are not creating the drivers to control how the pumps and meters function on the machines; we are creating a web application. Given the fact that there are many viable options to consider when building a web application, we have a large amount of freedom in how we design the architecture.

The most difficult issue for us, is that we are directly accessing the database holding Terrafarm module data. Since a different team is in charge of this, we will be relying on them to create an API that we can send HTTP queries to, to interface with the modules. We are then left relying so heavily on another team to create endpoints for user authentication, and other app-specific features. Because of this, we have chosen to have a database and API specific for the web application. This database will not try to replicate or compete with the other team's database. Our database will be holding statistics about the app's users, and how they are interacting with the site.

To keep things synced between the other team's database and our own, we will be storing 2 user IDs. One ID will be used to identify users of the web app, and one will be used to identify users of the terrafarm modules. It is much easier for us to build the authentication system with our database, and let the other team worry about core functionality. As users navigate around the UI of the web application, our server will be querying the other team's module database, to populate graphs and charts regarding the performance of the user's Terrafarm plants. We are creating a document of endpoints that will take X arguments, and return Y data in the specified format. For the first semester of this project, we will be using fake responses to mock responses from the other team's API. This way, we can complete the UI without having to rely on an outside service.

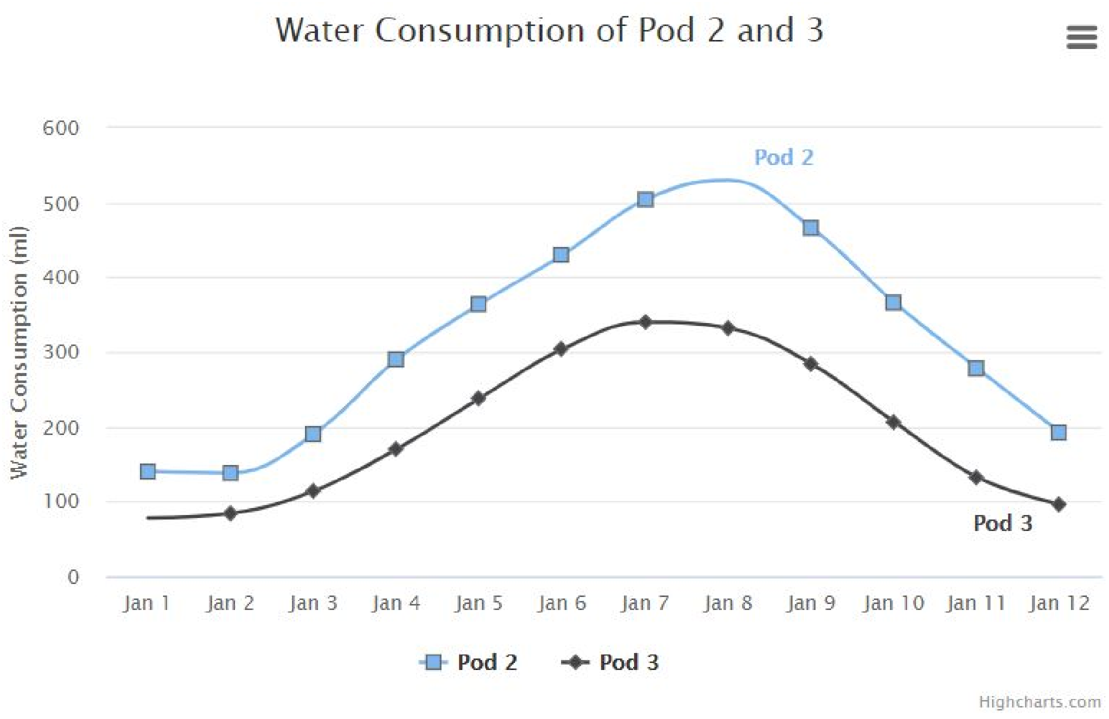
Another consideration that greatly determines how this app is designed, is how easy it will be to hire developers to work on Terrafarm in the future. Because of this, we wanted to choose a language and framework that will always be interesting to potential hires.

We have chosen to use Ruby on Rails as the framework for our web server. This server will be backed by a Postgresql database. The frontend of the Terrafarm web application will have some very complex UI elements, requiring something more robust than a simple templating language, such as ERB. We will be using React to accomplish these complex UI components. These languages and frameworks are very popular and updated regularly. There will be a mix of query types happening in our application. Some queries with be asynchronous, using the XHR protocol the receive data without reloading the web page. Other pages will be generated on the server, without the need for the web page to request further data.

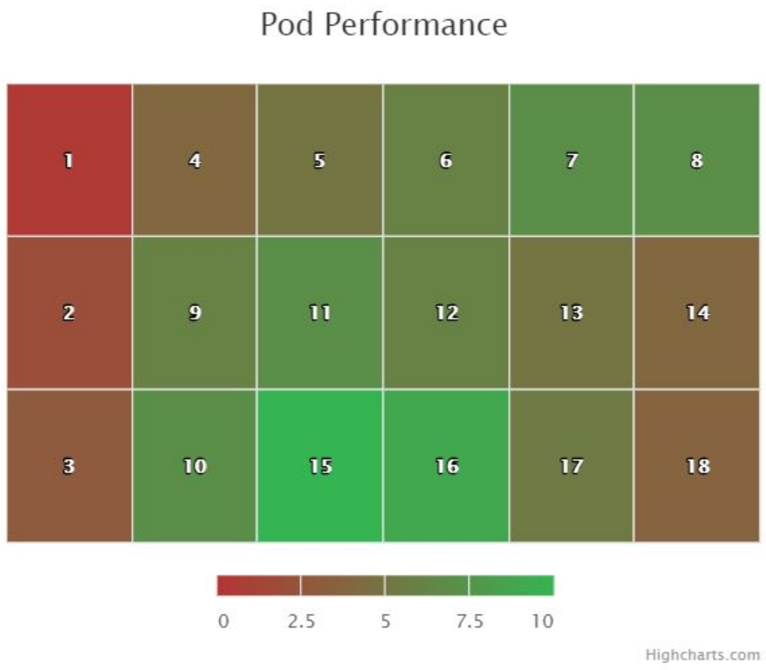
This team is comprised of computer scientists; we are not UI designers. Because of this reality, Frank (project owner) has found a designer who will be giving this team mockups to replicate. This takes a lot of pressure off of us in an area that we have no formal training, so we can focus on making this web application rich and complex.

The vision for the application is farther reaching than just a website however. The project owner has a vision for this to be a native mobile application as well. We believe that using service workers to create a progressive web application is a viable and future-proof way to accomplish both creating a website and a native application. Using this new technology, users will be able to add a shortcut to the app on their 'home screen,' and it will open in a window that does not feel like a browser (even though it really is).

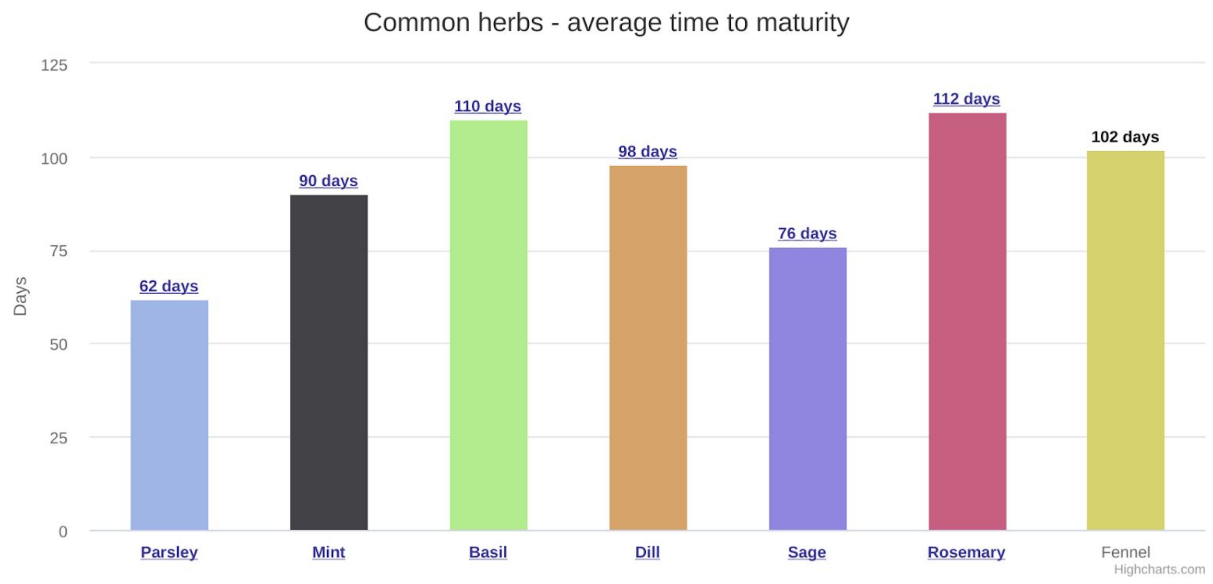
Data visualization is also something that we have to take into account. There are many good frameworks that assist in displaying data in chart and graph form, like D3, and Highcharts. We have decided exactly what we will do, but here is an example chart or water consumption of 2 pods over a period of 12 days using Highcharts.



The pods on each of the “drawers” will be individually monitored, and the web application users will have a view that displays how each of the pods is growing. We will likely use a grid of colors ranging from red to green to display how the quality of each pod. The image below is an example of this.



There is also an opportunity to help the customer decide what type of plant to grow. Below is an example of a chart that shows the time to maturity for various herbs.



The design constraints that we have to work with are mostly around how the user interface functions. Due to us working with an outside designer who will be telling us exactly what it should look like, we have extremely limited freedom when it comes to front-end structure. Again, the decision to use Ruby on Rails, Postgresql, and React could have looked much different, because of the sheer number of web frameworks that accomplish the same function.

Security is certainly another design constraint. We will be using the existing user authentication framework for Ruby on Rails, Devise. This gem is used widely in production applications around the world, and gives flexibility around how users are authenticated, and how frequently they need to be automatically logged out etc.

Ethical Issues

There are many ethical issues to consider in any software application and the Autofarm project is no different.

The primary concern in regards to the project as a whole is its effects on society and the environment. In theory, this product could help reduce carbon pollution by removing significant portions of the current supply chain for produce, help prevent topsoil loss and promote sustainable farming practices, and improve access to fresh, nutritious food in low-income areas that frequently lack it. It seems likely that Terrafarm could help in all of these capacities by reducing the average farm-to-table distance of many produce staples by growing closer to the places of highest demand, reducing the amount of topsoil depleting crops are grown in struggling fields by shifting the supply, and by allowing business owners and community activists in low-income neighborhoods to invest in the means to provide healthy food to their neighborhoods. On the other hand, the shift from classical farming methods could lead to unemployment and reduced income for traditional farmers and the people they employ, a group of people that are already struggling in general. Additionally, the initial cost of the Terrafarm modules may be too expensive to actually combat food deserts; however, as the costs of production go down as more units are sold and mass-produced, so too will the price, allowing greater access to this technology.

Intellectual Property Issues
Due to the nature of the project and the company's ultimate goal of making money from this product, we will have to be exceedingly careful that all of the software packages that we use are open source and allow for commercial use in order to be in compliance with intellectual property laws. Luckily, the main tools we are using are very generous with their licensing. Additionally, we need to be certain that our intellectual property in the form of any novel contributions to the website, application, and/or design elements thereof included in any portion of the final project utilizes are adequately protected in closed-source repositories.

Change Log
Initial Project Description -> Project Proposal

- The title, synopsis, and description were all slightly modified to reflect the increased scope of our project. In addition to building the Autofarm website, we are now also building the Autofarm mobile application in the form of a progressive web app.
- The project milestones were altered to reflect the plan created with the Autofarm directors. Our first semester will focus on development of the website, while the second semester will include development of the mobile app and integration with other Autofarm systems.
- A section of the work plan addressing the need for a liaison to Autofarm leadership/groups was removed, as all the Autofarm teams and directors are now in contact through Samepage.

Project Proposal -> Final Project Design

- Removed the word "website" from the project synopsis, as we've decided to fully focus on creating a progressive web application
- Removed milestone #3 for first semester, which stated "Finish minimum viable product of website", for the same reason
- Added estimated completion dates for each of the project milestones for the second semester
- Updated the Gantt chart to show the updated progress on the project