Aerobicon Partnership

Team Members:

Shane McClain, Justin Carnes, Cody Davis, Josh McConnell,

Maison Abmas, Megan Skorzewski

Abstract

This report is intended to explain the Aerobicon project in its entirety. We intend to provide insight to our collective vision for the device's functionality, marketing, and customer base as well as our own corporate structure, our data backup philosophies, our required backend services for data management, the setup and installation of the device, the software design and development process, and our plans for the future.

Table of Contents

List of Figures

List of Tables

Executive Summary

Product

Aerobicon manufactures an internet enabled device for monitoring and controlling aerobic septic systems. Our product is a reliable, easy-to-use device that property owners can use to easily monitor the state of their septic system, from virtually anywhere.

Customers

The target market for the Aerobicon device are current owners of aerobic septic systems looking for more control of their system, new homeowners that may be considering installing an aerobic system, and licensed septic system installers.

Market

The Aerobicon device is designed to compete with current manufacturers of simple timing circuits and control panels for aerobic septic systems. We believe in our product and believe that we can offer a better alternative for controlling aerobic systems.

Company Description

Mission Statement

To provide a reliable, high-quality product that gives property owners more time to focus on what's important to them, and less time on property maintenance.

Principal Members

Shane McClain – President and Lead Developer

Justin Carnes – Co-Developer and Visionary

Maison Abmas – Technical Engineer and Visionary

Megan Skorzewski – Technical Engineer and Visionary

Cody Davis – Systems Analyst and Visionary

Josh McConnell – Systems Analyst and Visionary

Legal Structure

Aerobicon is a Limited Liability Company, founded in Tyler, Texas.

Market Research

Industry

Aerobicon will be part of the septic system controller industry. Currently, there are multiple manufacturers of septic system timers, alarms, and control panels, all of which serve only one (or few) simple functions. Aerobic septic systems treat raw sewage by chlorination, then utilize a combination of water and air pumps to spray treated water into a drain field. Where older septic systems consist of a single tank that must be emptied periodically by a licensed septic system company, aerobic systems are more complex, and require more routine maintenance, but offer a more environmentally friendly solution, and can be installed in places where soil isn't suitable for a traditional septic tank.

Our goal is to penetrate as much of this new market as possible, while also converting current homeowners to our device, which offers more control over the monitoring and scheduling of maintenance on these systems.

Customer Segments

Aerobicon will provide its device to a multi-sided market. The targeted end user segment is:

- Current and future homeowners looking to upgrade their existing aerobic septic system to make use of the monitoring capabilities and user control that our device provides
- Homeowners needing to replace their existing septic system
- New home builders (the TCEQ estimates that 20% of new homes built in Texas utilize an aerobic septic system)[1]

Our target customer segment is septic system providers. Due to the regulations regarding installation of septic systems, the Aerobicon device will be sold exclusively to this customer segment. Service for each device will be provided to both sides of the target market; septic system installers will be able to contact our customer support for help in connecting the Aerobicon device to the septic system, while homeowners will be able to get support with the website, network configuration, email alerts, or setting the system's timers.

Distribution Channels

Following our market strategy, the Aerobicon device will be sold in a level 1 selective distribution strategy: Aerobicon will sell the device directly to licensed septic system dealers, who in turn sell to the end user at a markup that will provide adequate profit margins to both the septic installer and Aerobicon.

[1] https://www.tceq.texas.gov/assistance/water/fyiossfs.html

Product Advantages

Aerobicon is an internet enabled monitoring device for your septic system. With Aerobicon, you can replace more costly timers and tank alert systems with a single device that will alert you via email when tank levels are too high. Our easy to use Web interface allows you to adjust the spray timer or set your own specific time of day to activate the sprayers on your septic system. The benefits provided by Aerobicon include:

- Web based interface allows you to set specific timers or set specific days and times that your septic system's sprayers are activated.
- Receive email alerts when water levels are too high/low, allowing you to delay or start sprinkler heads automatically and remotely.
- Retrieves local weather data to best determine when to activate the sprinkler.
- Compatible with new or existing aerobic septic systems.

Service Line

Product/Service

Aerobicon will sell the primary device, which is based on the Raspberry Pi architecture. In addition, each device will come included with a 90-day limited warranty, with the option for the end customer to purchase an extended warranty as an annual subscription. We will also offer post-installation technical support, at an hourly rate.

Revenue Streams

Aerobicon will rely on the following three revenue streams:

- Transaction Revenue – Direct sale of each Aerobicon device at a price of $200.00; the Aerobicon unit will be sold directly to licensed septic system installers. Suggested retail price (SRP) of each unit will be $300.00, a 50% gross profit margin for the reseller, and comparable to similar septic system control panel units currently in the market.[2][3]
- Service Revenue – Aerobicon will offer post-installation technical service to the end user at a rate of $50.00 per hour; this rate is well below current hourly rates for similar support in the East Texas region.
- Recurring Revenue – The Aerobicon device will include a 90-day limited warranty, with the option for the end user to purchase an annual maintenance plan for $50.00.

---

[2] Pricing for a similar product was sourced from:
https://www.septicsolutions.com/septic-parts/control-panels/aerobic-control-panels/rab202-fs_rab-202-outdoor-aerobic-alarm-and-control-panel?gclid=CjwKCAjw1cX0BRBmEiwAy9tKHmmSVkd3Zkdu_4zW4pBFiqy_YkB6Fog0JlY0SgHty3WJftrZjuTA5xoCYfIQAvD_BwE

[3] Pricing for another similar product was sourced from:
https://www.wholesalesepticsupply.com/products/acp-dmd-2a-dual-light-septic-control-panel?utm_source=Google+Shopping&utm_medium=Shopping+Campaign&utm_campaign=PLA&variant=20478376935512&sfdr_ptcid=20902_4_482707103&sfdr_hash=4596cb00179dfa518734b25d7cef4a83&keyword=&device=c&gclid=CjwKCAjw1cX0BRBmEiwAy9tKHll2EBuriiVXI8igK5AM0W1ZcOlgmvH7gAsFfxyQpYSfwZRDLss2ERoC4wEQAvD_BwE

Cost Structure

An analysis of our cost structure is provided below. While economies of scale may ultimately influence the number of break-even units sold, we do not predict significant cost savings during the first year of operation. However, this may change as we secure partnerships with our hardware manufacturers. Our initial plan is to operate with minimal staff; this will help keep costs at a minimum and staying within our initial sales territory of East Texas will not only simplify operations, but also keep the costs of delivering products low.

Calculations in the cost structure were based on the following factors and assumptions:

- Revenue per unit of $200.00 (see the section *Revenue Streams* for more information)
- Minimum sales of 100 units per month
- Our multi-channel distribution model

Variable Costs:

Particle Photon WiFi Development Board	$19.00
Relay Switches[4]	$4.99
Resistors	$0.96
Charlotte Pipe 1-1/2-in dia 330 PSI PVC Pipe[5]	$5.73
Charlotte Pipe 1-in dia x 10-ft L 200 PSI SDR 21 PVC Pipe[6]	$2.60
LASCO 1-1/2-in x 1-1/2-in dia Cap PVC Fitting[7]	$1.14
Cylewet 10Pcs Reed Switch Normally Open (N/O) Magnetic Induction Switch Electromagnetic for Arduino (Pack of 10) CYT1004 [8]	$2.40
Power Gear Telephone Line Cord, 100 Feet, Phone Cord	$4.79
Monoprice RJ11 6P4C Plug Flat Stranded, 50-Piece/Bag (107268) [9]	$0.37
Orbit 2 Pack Plastic Grass Over Growth Sprinkler Guard Donut (2 halves)[10]	$9.99

[4] Product sourced from: https://store.particle.io/products/photon?variant=9097143491

[5] Product sourced from: https://www.lowes.com/pd/Charlotte-Pipe-1-1-2-in-x-10-ft-330-Sch-40-Solidcore-PVC-DWV-Pipe/3133037

[6] Product sourced from: https://www.lowes.com/pd/Charlotte-Pipe-1-in-dia-x-10-ft-L-200-PSI-SDR-21-PVC-Pipe/1000080801

[7] Product sourced from: https://www.lowes.com/pd/LASCO-1-1-2-in-x-1-1-2-in-dia-Cap-PVC-Fitting/3371576

[8] Product sourced from: https://www.amazon.com/gp/product/B01NBPDU04/ref=ppx_yo_dt_b_asin_title_o04_s01?ie=UTF8&psc=1

[9] Product sourced from: https://www.amazon.com/Power-Gear-Modular-Machine-27638/dp/B07P92LKZS/ref=sr_1_7?dchild=1&keywords=flat+phone+wire&qid=1587528658&s=electronics&sr=1-7

[10] Product sourced from: https://www.amazon.com/Monoprice-RJ11-Stranded-50-Piece-107268/dp/B0069LVX1Y/ref=sr_1_22?dchild=1&keywords=telephone+connectors&qid=1587529926&s=electronics&sr=1-22

	Neodymium Disc Countersunk Hole Magnets with 10 Screws, 1.26 inch D x 0.2 inch H Strong Rare Earth Magnets - Pack of 10	$3.00
Parts & Materials/unit:		**$54.97**
Delivery/order		$35.00
Utilities/mo.	Electric	$174.00
	Water	$55.75
TOTAL:		$319.72
Variable Cost/Unit(100 Units)		**$3.20**
Revenue/Unit		$200.00
Variable Cost/Unit		$3.20
Contribution Margin/Unit		$196.80
Break Even Point(Units)		**2,812**

Table A1

Marketing and Sales

Growth Strategy

We plan to start with a lean operation, with eight employees, partnering exclusively with licensed septic system installers in East Texas. Our goal is that through steady sales, we can look at expanding into other territories, beginning in Texas with possible expansion into other states, within five years.

Customer Communication

Aerobicon will communicate with its customer base by:

- Direct calls and face-to-face meetings with licensed septic system installers within our sales territory
- Providing our company phone number and email address on our company website, Facebook, and Twitter pages.

How to Sell

Aerobicon currently consists of six employees. Although we are in separate roles, we are all confident in our ability and position as company founders to individually market our product and services. As sales revenue increases, we plan to hire at least two additional employees to act as our territorial sales manager and marketing director.

Aerobicon Backup Plan

This document will list what to do in case of certain situations that may come up. Please keep in mind that every situation is unique and depending on the circumstances it might be necessary to do things differently.

Full Backups

Routine Backups

- Every 3 months, a full backup will be done on both the Google Drive and the hard drive.
- This includes all customer information, financial information, sales information,employee information code, etc.
- As the company grows we will move to having different hard drives for the different types of data, but for now we will have only one.

Anticipated Catastrophic Event

- In the anticipation of a catastrophic event, a full backup should be done on Google Drive and the hard drive.
- Employees should also make sure to have all changes saved on their OneDrives prior to the start of the Google Drive and hard drive.

Incremental Backups

In Case of Code Adjustments-On Site Backup

- The CEO and CTO and whoever they choose to add will have access to a folder on Google Drive.
- In order to change the code on file one of them will need to log into Google Drive
- Click on the “Aerobicon” folder to open it
- Select the file that needs to be updated
- Save the old file under “Aerobicon_Version_x”(x being the number of the version)
- Be sure to include any other important indicators of the file’s name
- Open the file and make any necessary changes to the code
- Save the changes to the file and upload it to the Google Drive

In Case of Server Failure-On Site Backup

- In the Aerobicon Google Drive there will be a folder titled "Aerobicon Server"
- All of the server files are backed up into this folder
- Open the "Aerobicon Server" folder
- Download the necessary files (dependent on the reason of the crash)
- Click on the "Server History" document and log the date, incident, and incident solution.
- Save your changes and reupload the file.
    - Be sure to identify the server's problem area(s) if it was responsible for the crash
    - Check the network issues
    - Maintain detailed records of why the server failed and how the problem was resolved.

In Case of Code Adjustments- Off Site Backup

- The CEO and CTO will decide on the location of a hard drive containing all of the information that is stored in the Google Drive folder. This hard drive will be updated annual
- Click on the Aerobicon Code Folder
- Open the necessary file and make changes
- Save your changes as a new file and upload the new file
- Eject the hard drive and return it to its location

Employee Data

- Data that employees use on a regular basis will be stored on the employee's computer using OneDrive. Once a quarter all the data will be backed up to Google Drive and the external offsite hard drive.
- All employees will have access to the OneDrive, the employees that need it will have access to the Google Drive.

In Case of Catastrophe

- If connection to Internet is possible:
  - Login to Google Drive to gain access to the files that need to be recovered
  - Download/install the necessary files
  - Make sure to document the incident and the solution
- If connection to the Internet is not possible:
  - Locate the external hard drive
  - Plug it into a computer and enter the password to gain access
  - Access the files needed
  - Be sure to document the incident and the solution

Testing Backups

When to Test

- Backup tests will take place every 6 months and when necessary
- Tests will be scheduled and carried out as efficiently as possible
- Tests can also take place during any of the following circumstances:
    - Significant application change
    - New application being built
    - Application data changed
    - Catastrophic event has occurred
    - Test is ordered by CEO or CTO

What to Test

- File Recovery
    - Test to make sure that files are being recovered correctly
    - Example: Open files on the recovered device and compare them to the files opened on the normal device
- Physical Recovery
    - Test to ensure that if necessary servers and computers can be reconfigured using backed up data
    - Example: Use the backed up data to set up a company server and or computer
- Data Recovery
    - Test recovered company database in search of any errors
    - Example: Create a mock catastrophic incident in an isolated environment and do a complete recovery

Aerobicon Backend Services

The transfer of data used in Aerobicon is very unique. The data will be handled by using the client's Node.js server that runs on the Raspberry Pi. The Raspberry Pi will host a local webpage that will allow the user to set and update their settings and view their past and current log files. The server will also be using the GPIO pins to scan for input from the float switches, queries from the Open Weather API when applicable, activation of the pump, and recording of activations to a log file. At this time, we plan on storing the user's settings and log files under a locked admin file on the Raspberry Pi.

The admin account will be set up on the webpage within the server. Once this is done the client will be able to make edits to their settings or any other services such as the float switches or pump activation. These user specific settings will be stored in the webpage that is hosted by the Node.js server and gets stored in a file on the Raspberry Pi. It will be stored in the Node.js server directory named "UserSettings.json". The settings include the client's desired activation time, the city and state that the aerobic septic system is located in order to accurately pull weather data, and the user's name and email that will be used for alerts such as the tank being too full. When the pumps are activated they will trigger a time stamped log file that will be stored as a text file in the Node.js server directory.

Septic system input will be in the form of on and off signals from the multiple float switches that are installed in the tanks. These float switches indicate the water level of the second and third tanks. If a tank sends an “on” signal, it means that the water level has reached the float switch. Each septic system can have a maximum four switches total with two switches per tank. One switch will indicate that the tank is full or almost full and needs to be pumped or that the tank has not been pumping properly and needs maintenance. This input will be transmitted to the I/O pins on the Raspberry Pi and read by the Node.js server that is constantly running on the Raspberry Pi.

In conclusion, the backend services for Aerobicon rely heavily on the Raspberry Pi and the Node.js server. This is because it is the most effective way to handle all of the data.

Aerobicon Tech Needs

Aerobicon is an IoT device that links up with your septic system and allows you the ability of monitoring your system remotely. We currently use a Raspberry Pi that is running Raspbian OS and Node.js along with float switches within the tanks. The following is a list of things needed that we currently do not obtain before we would be able to go live. Since there is no battery source needed, an added power out for the Raspberry Pi will be required.

WordPress Design

Our website was designed through the use of WordPress. We chose this route because it gave us complete control of our website's design and layout therefore, we were able to create a custom website and obtain complete control of each aspect within it.

Amazon Web Services Website Hosting

We decided to host our website on AWS. We went this route because it is the most effective, user friendly option and also has multiple elements that will be helpful as our company continues to prosper.

Amazon Web Services Database System

As we continue to expand, we are planning to use Amazon Relational Database Services, more specifically Amazon Aurora. We chose Amazon Aurora due to us liking its ability to combine availability and affordability. In addition, it is faster than PostgreSQL and MySQL and, majority of the administrative maintenance will be done through Amazon Relational Database Services.

Amazon Web Services Cloud

To begin, we will only be doing backups through Google and then branch out to AWS Cloud Database for backups as our company expands. This could either be in addition to the Google Drive or even completely replace it.

Google Drive

For our backups we will be using Google Drive therefore, we will need to purchase storage space through Google. Some things that we plan on backing up will be employee information, product information, code, reports, etc.

Hardware

The hardware needed will be specific to the users exact septic system. We will need a power outlet that we can branch off of the systems powerline, along with relays that we can connect to the systems float switches and Raspberry Pi. Finally, an ethernet cable will be needed for installation but, after installation the system will use the consumers WIFI network.

PayPal

We will use PayPal in order to ensure security of payments in pair with them only charging a small fee of 2.9% for each transaction. PayPal also allows us to send and receive payments in person and online as well as invoices.

Shopify

For our online storefront we will be using ecommerce platform Shopify. This platform is commonly used within small businesses due to it being flexible, affordable, and having a user-friendly design. Shopify also ensures consistency for all users no matter the device they are running Aerobicon from while allowing the ability to personalize it to our certain wants and needs and provides room for growth as we expand.

Conclusion

In conclusion, the items mentioned throughout the document are what we foresee us needing before Aerobicon is able to go live. As our company continues to expand, we realize additional resources will be required. This document is a good guide of the necessary services needed before we are able to launch Aerobicon and will be used as a way to assess our growth over time.

Aerobicon User Manual

Overview

The following user manual is a guide on how to set up and install the Aerobicon Septic System in your home. This guide will be broken down into 3 major sections for the user's convenience with the Installation section being broken down into subsections based on the user's septic system set up.

Table of Contents

Materials Needed for Setup

Aerobicon Septic System kit (includes the following):

- Aerobicon device
    - AC Adapter
- Aerobicon Float Sensor or Floats (4)
- Wall Socket

Aerobicon General Setup

1. Open your package and remove the Aerobicon Device and Adapter.
2. Plug the adapter into the Aerobicon Device to power it on.
3. Connect the Aerobicon Device to the internet by downloading the Photon application on your mobile device or computer in order to connect to wifi.
4. Next, from any web browser, travel to aerobicon:3000. Once the page loads, you will be able to register on the Aerobicon web portal, and then sign in.
5. Once you are signed in, click settings, and set your settings appropriately to begin using the device.
6. At your convenience, you can change your settings, test your pump, and view logs, provided that you are signed into our website.

Installation

<u>* With the purchase of our float sensor, you can directly attach it to the Aerobicon device. If you do not want to purchase our float sensor and want to use 4 floats, you will have to buy an adapter for our device and wire the floats to the adapter.*</u>

<u>*To be completed after a professional has come to install a new socket.*</u>

<u>*Special care is needed when disconnecting floats from the tank. Please do NOT remove floats if you do not have professional experience.*</u>

1. Have a professional "pull" wire for a wall outlet to be installed inside of the box.
2. Turn off the power.
3. Plug in the power adapter for the device.
4. If you have any floats connected to the pump's main circuit, then you can either remove them and rejoin the circuit, or you can raise them up and zip tie them to keep them on.
5. If you have any floats not connected to the pump's main circuit and not in the second tank, then ideally, they need to be removed.
6. If there is one float in the second tank, then you can disconnect it from the circuit it's attached to, and use it with the Aerobicon device by connecting it to the float sensor's adaptor or connect it to the float's adapter to be used. Otherwise, you will need another float for this tank.
7. Connect the float sensor's adapter to the Aerobicon device, or use the float's adapter to connect 4 floats to the device.
8. Turn on power.
9. Download the phone app "Photon" for the device.

10. Using the app, connect the device to the internet.

11. Connect to Aerobicon:3000 and set and save your settings.

12. Test the pump.

Aerobicon SDLC

Scope

Aerobicon has been developed with the primary goal of replacing a traditional timing circuit for an aerobic septic system. To this end, we have pursued the tasks relevant to fulfill this goal, as well as any additional quality of life features that fit within our development timeframe. The features that we chose to include are full pump scheduling functionality, weather-respondent pump activation, viewable activity logs detailing pump activation history, the ability to test a pump's functionality, and email notifications when tanks get too full. These features and their associated settings are served to the user through a web portal, locally hosted by the Aerobicon device.

Technology, Hardware and Software

The current iteration of the Aerobicon project depends upon the following technologies. Please note that this list, particularly hardware, is set to change before product launch (see **Planned Features**, Page 55):

1. **Raspberry Pi** - The brains of the Aerobicon. The Pi is responsible for running all software between the user and the septic system in our prototype. The Pi contains the following sub-components:
    a. An operating system - Raspbian OS.
    b. SD Card - The Pi uses this as its storage mechanism. OS and all files are stored here.
    c. A Climate Controlled Box - Keeps the Pi and other components insulated from extreme weather and temperatures.
        i. Existing septic systems already place their controls in these boxes. We expect that customers will have one currently installed, therefore this is not something we will provide.
2. **Wifi** - As the device is controlled via a web page, it requires an Internet connection.
    a. The user will initially need to connect the Pi to the Internet via an ethernet cable. This initial connection will allow the user to access the setup page and configure a Wifi connection.
3. **Openweather API** - Our software calls this API to manage weather settings.
    a. With this data, the user is able to set the sprinklers to run only when the tank is full enough, their preset time has come to pass, and it is not currently raining.

4. **Node.js** - This is the framework through which we serve the user their device's settings page and execute scripts.

Functional Requirements

Functionally speaking, Aerobicon must:

1. Provide a web-based GUI.
2. Allow the user to modify settings.
3. Keep a record of pump activations.
4. Report log data upon request from the user.

Non-Functional Requirements

Other requirements for Aerobicon include:

1. 24/7 operation and accessibility.

Lifecycle model

We developed this project in accordance with the iterative lifecycle model. This allowed us to ensure components were completed and working together properly before we moved on to polish and add extra features. Once we have launched the device on the market, we will support it by improving existing features and adding new features, according to user feedback. This support phase will continue indefinitely.

Graphic Use Case Model

Figure A1 illustrates the interactions between the three major actors in the Aerobicon system: the user, the Aerobicon device (consisting of the Pi and its software components), and the septic system (consisting of tanks and float switches). The user is kept blind from the inner workings of the other two actors, and the Aerobicon is kept blind from the inner workings of the septic system. The user's sole interaction is with the Aerobicon, and the Aerobicon mediates between user input and I/O provided by the septic system's float sensor. The functions described by each oval in the diagram show the actions performed between the initiating and performing actors, and the arrows indicate the flow of data during the actions.

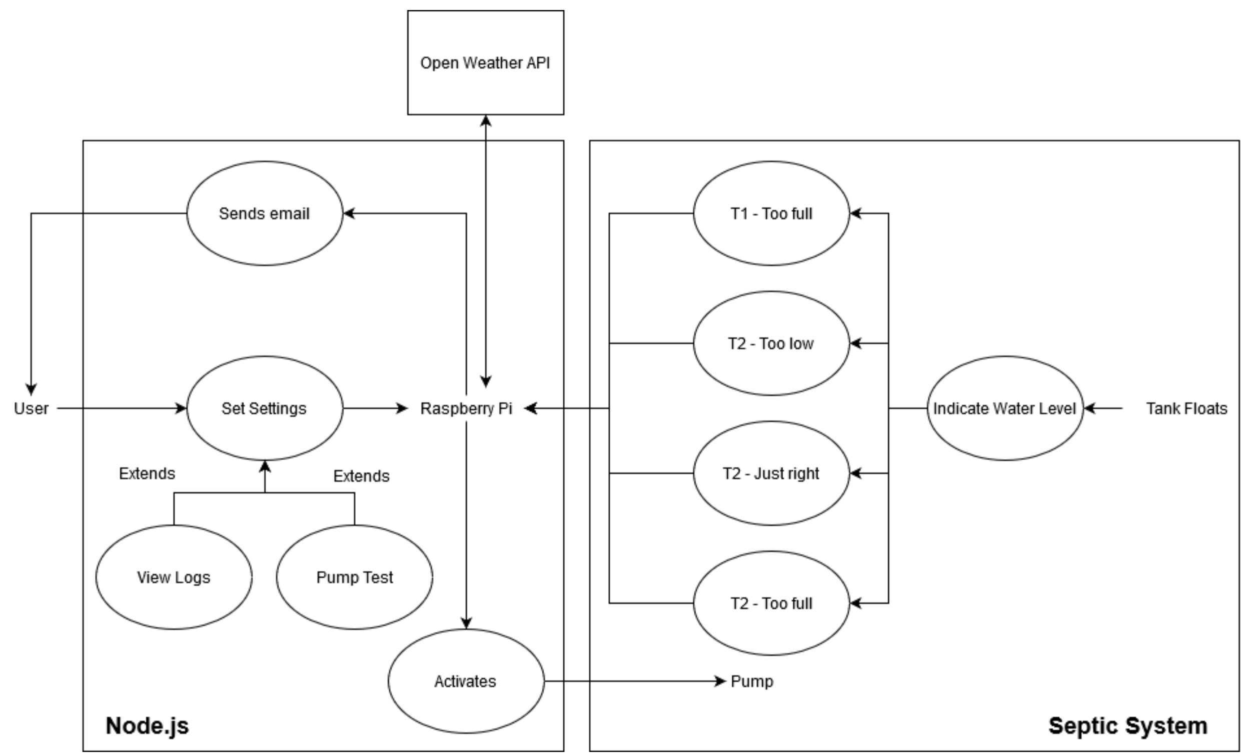

Figure A1

Textual Description

Primary Use Case

Set Settings:

- Participating Actor: User
- Entry Conditions: User navigates to their device's webpage.
- Flow of Events: User interacts with the page and controls device settings.
- Exit conditions: User saves their settings and closes the page.

View Logs:

- Participating Actor: User
- Entry Conditions: User navigates to their device's webpage and requests to view their device's logs.
- Flow of Events: User is given a log of the pump's activity.
- Exit conditions: User closes the log and the webpage.

Sends Email:

- Participating Actor: Raspberry Pi
- Entry Conditions: Any tank's top float switch is triggered.
- Flow of Events: A script is executed on the Raspberry Pi, sending an email to the user that a tank is too full.
- Exit conditions: The email script finishes executing.

Pump Test:

- Participating Actor: User
- Entry Conditions: The user navigates to their device's webpage and schedules or triggers a pump test.
- Flow of Events: The user requests a pump test, and the Pi activates the pump.
- Exit conditions: The user stops the pump test, or the predetermined test time limit is exceeded.

Activates:

- Participating Actor: Pump and Raspberry Pi
- Entry Conditions: The Raspberry Pi's activation criteria are met.
- Flow of Events: The Raspberry Pi executes a script that starts the pump.
- Exit conditions: The tank empties below the second float switch, or the preset run time limit is exceeded.

Activity Diagram

Figure A2 below illustrates the flow of a user's interaction with Aerobicon. Access to the user's webportal is gated behind a username and password. Once logged in, users will be on the portal's homepage, where they have the ability to view/change settings, view logs, or test the pump. If a user sits inactive on a page for 5 minutes, they will be logged out and sent back to the login page.

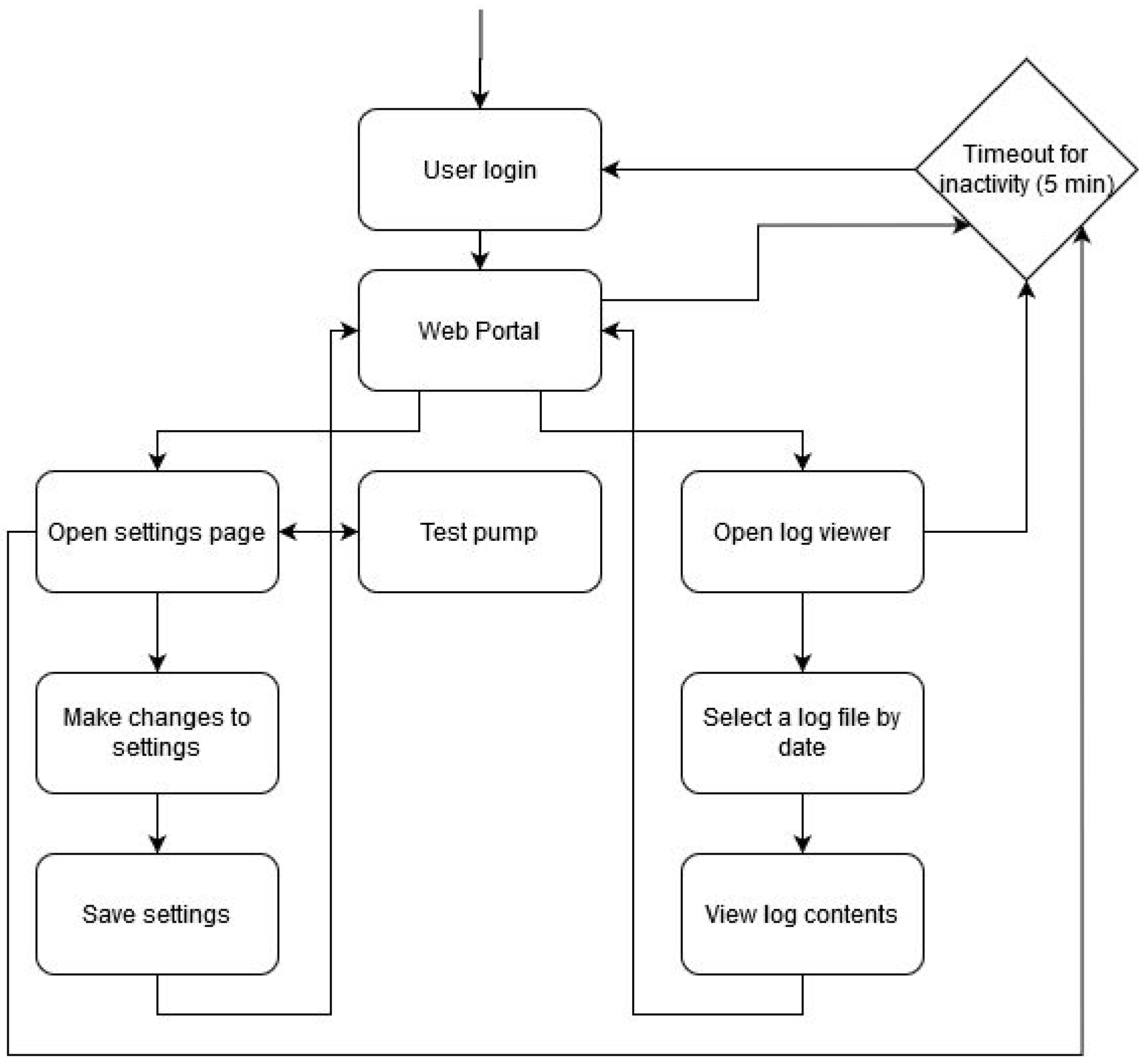

Figure A2

Sequence Diagram

Below, **Figure A3** outlines the sequence of events beginning with user interaction with Aerobicon, and ending with Aerobicon's reactionary steps.

In the main scenario, the user accesses their landing page, clicks on the "View Settings" link, and sets their desired system settings. The Node.js server running on the device then saves these settings to a JSON file and sets new cron jobs based on the changes. Meanwhile, the server is also polling the overflow switches in the tanks every 15 minutes to ensure that water is not backing up- if these switches are active, the pump is activated immediately and the owner is notified via email. When the time specified by the user for pump activation arrives, the device checks to see if the minimum fill level switches are activated and if the weather is clear. If these conditions are met, the pump is activated, thus setting off the sprinkler system.

In the alternate scenario, the user navigates from their landing page to the log viewer. They then navigate through the folder hierarchy and select the log they would like to view. The Node server loads the requested log from the device's memory and serves the contents to the user.

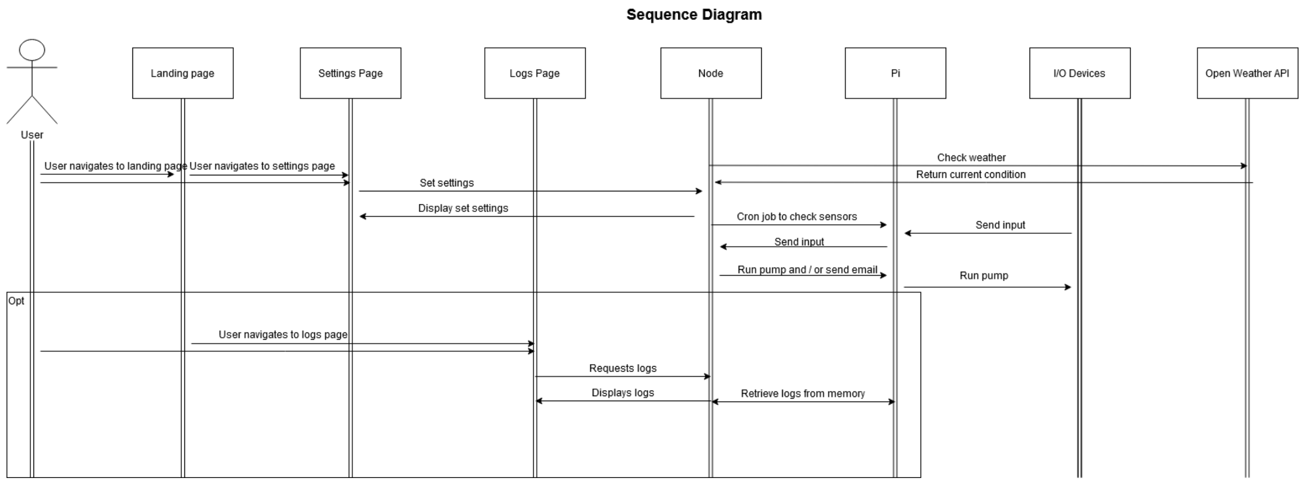
Sequence Diagram
User
Landing page
Settings Page
Logs Page
Node
Pi
I/O Devices
Open Weather API
User navigates to landing page
User navigates to settings page
Set settings
Display set settings
Check weather
Return current condition
Cron job to check sensors
Send input
Send input
Run pump and / or send email
Run pump
Opt
User navigates to logs page
Requests logs
Displays logs
Retrieve logs from memory

Figure A3

Architectural Design

The main Node.js program consists of 4 components:

- The User Settings modules,
- the Weather API interface,
- the scripts to carry out different calls to and from hardware,
- and the user-facing web portal

Through the use of the Express.js library, Aerobicon hosts a local web server that serves the user with three distinct pages: a landing page, a settings page, and a log-viewer. On the settings page, the user can control the scheduling of the pump, configure weather preferences, provide an email address if they wish to receive email alerts, and test fire the pump. Once the user modifies the device's settings, Node.js sets a cron job to ensure events occur in accordance with the new settings.

We designed our software package to follow a "Main Program/Subprogram" architecture. Our vision of having Node.js handle the communication between the user, hardware, and various scripts turned out to translate very cleanly into our development of the device. The planned architecture we followed is modeled below in **Figure A4**.

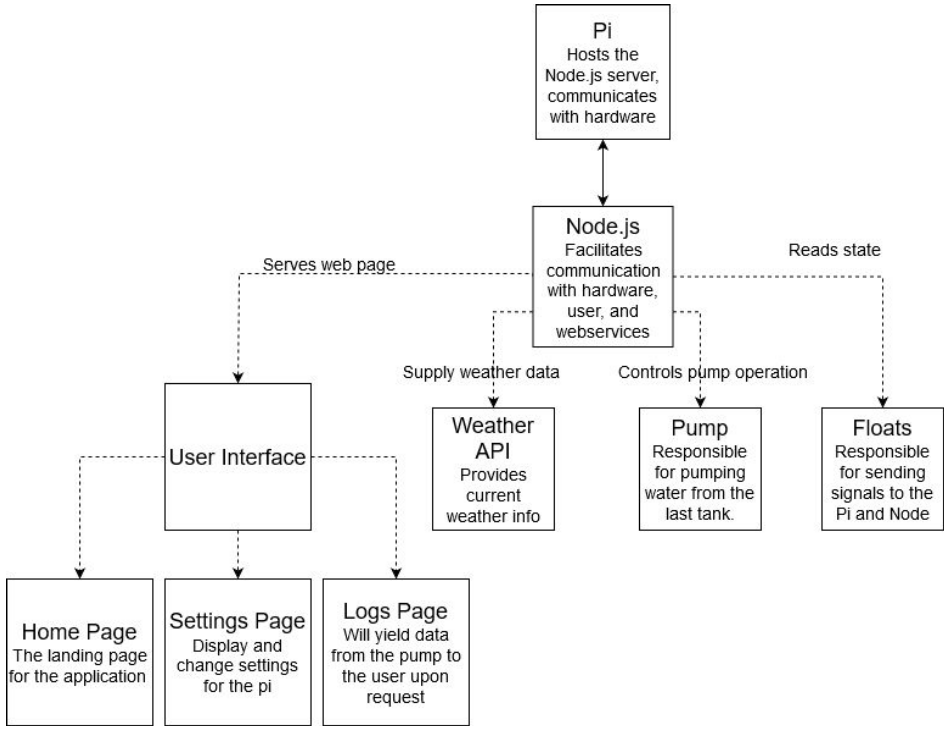
Pi
Hosts the Node.js server, communicates with hardware
Node.js
Facilitates communication with hardware, user, and webservices
Serves web page
Reads state
Supply weather data
Controls pump operation
User Interface
Weather API
Provides current weather info
Pump
Responsible for pumping water from the last tank.
Floats
Responsible for sending signals to the Pi and Node
Home Page
The landing page for the application
Settings Page
Display and change settings for the pi
Logs Page
Will yield data from the pump to the user upon request

Figure A4

User Interface

Design Philosophy

When envisioning the design for the user-facing web portal, we knew that a simple approach would be the most effective. The features of the web portal are not complex, therefore it would not make sense to wrap them in a confusing interface.

Control of the Aerobicon device occurs through three distinct webpages. The first page is a homepage, containing links to two other pages. One such page is the settings page, where users can view and control settings. The other is a log viewer, where users can retrieve and view the contents of activation logs.

Landing Page

Figure B1 shows the landing page for the user's personal web portal. This page is unique to each Aerobicon device, and contains links to the settings page and the log viewer. Users can access this page via the url "aerobicon:3000". Because this is a local server, we are able to use the device's hostname and port rather than a true web domain.

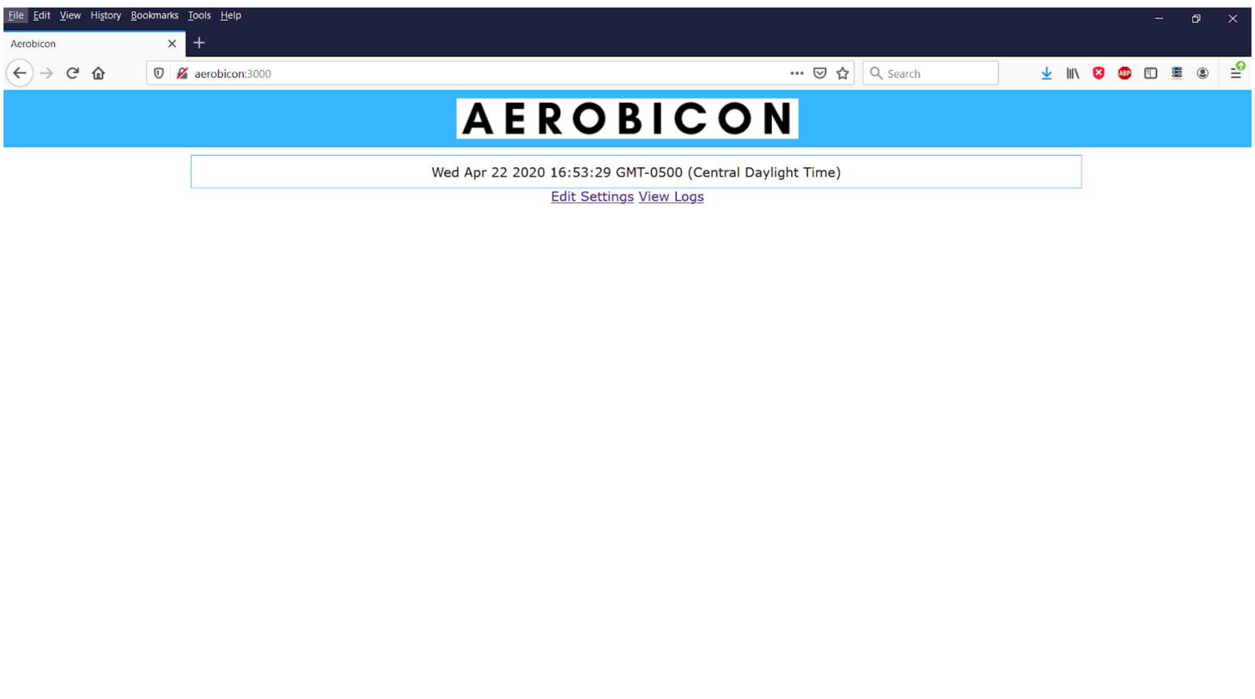

Figure B1

Settings Page

Figure B2 shows the device's settings page. This page allows the user to provide their name and email (in order to receive email alerts), set the time at which they would like the pump to activate (provided other activation criteria such as weather and water levels in the pump are satisfied), and configure their location information required to retrieve weather data. There is also a button labeled "test pump", which activates the pump for a short time to ensure that it is functioning properly.

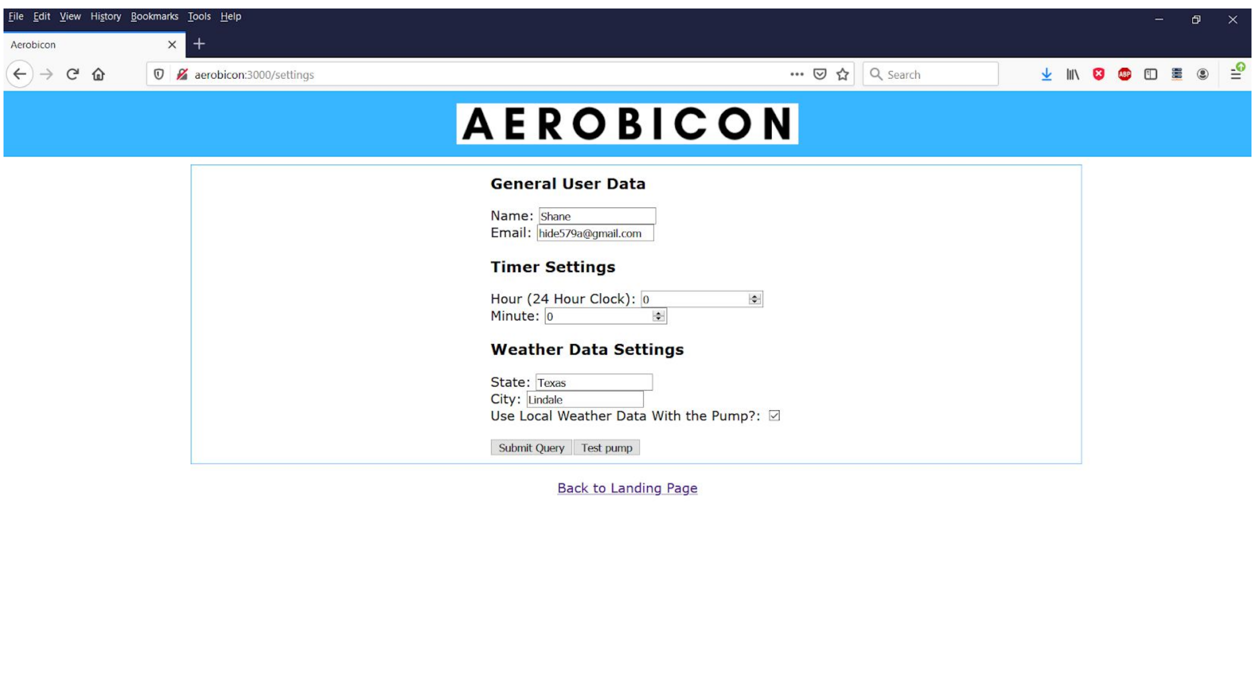

Figure B2

Logs Page

Figure B3 shows the log viewer page. This page allows the user to select a date and view any logged activation data for that date. The logs are stored hierarchically in folders by year, month, and day. The user can move through this hierarchy and select a log from a specific day of the year. When the user selects the day, the contents of that day's log file are displayed on the page, as shown in **Figure B4**. This information is useful for troubleshooting purposes- if the pump was scheduled to run but did not, the owner can view the logs to see if the activation was rescheduled due to weather, or if the device told the pump to run but it did not, indicating hardware failure.

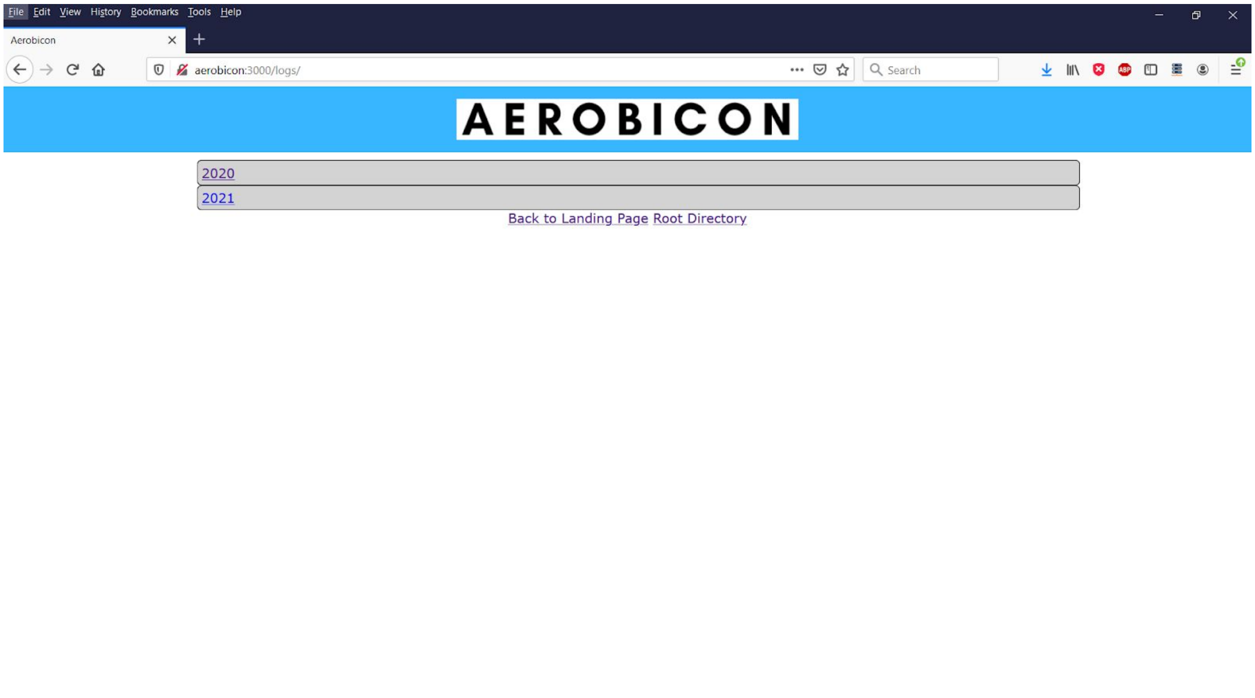
File Edit View History Bookmarks Tools Help
Aerobicon
aerobicon:3000/logs/
Search
AEROBICON
2020
2021
Back to Landing Page Root Directory

Figure B3

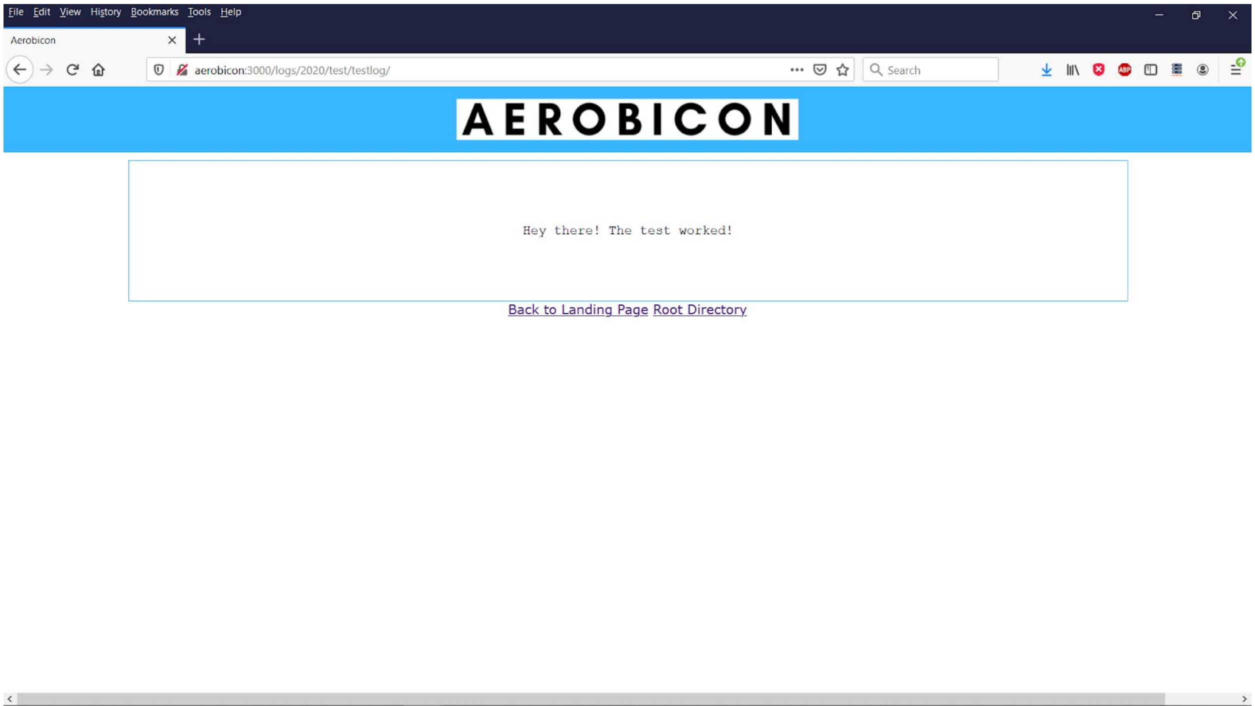
File Edit View History Bookmarks Tools Help
Aerobicon
aerobicon:3000/logs/2020/test/testlog/
Search
AEROBICON
Hey there! The test worked!
Back to Landing Page Root Directory

Figure B4

Testing

The parts that comprise our prototype device are as follows:

- The Raspberry Pi and all required software:
    - Raspbian OS
    - Node.js with important packages such as OnOff and Express.js
    - The Aerobicon software package, including user-facing web portal and scripts to trigger pump activation, retrieve weather data, write logs, etc.
- A mouse, keyboard, and monitor
    - Software packages are installed to allow for "headless" operation as well, through remote desktop software
- A wireless network connection
- Float Sensor
    - This is a piece of circuitry that the team is developing, consisting of magnetic reed switches and PVC pipe, among other components. It will make the installation process smoother and cheaper, and will allow for maintainers of the system to easily adjust the heights of the floats if need be.

This prototype has allowed us to test the software integrity, making sure that the different scripts work as intended and the logic is in place for interpreting input from float switches. Our next big step for this project is to transition to our final hardware iteration. The parts for this version of the device are as follows:

- A Particle Photon microcontroller and all required software
- A wireless network connection
- An aerobic septic system
    - The system will need to be modified to house the device: power will need to be routed to the Aerobicon device, and the system's float sensor will need to be installed in the tanks and connected to the device via relays.

Unit Testing

This step was handled entirely on the device itself. Because Aerobicon is built on a Raspberry Pi, it has a fully functioning desktop interface. We made use of the device's own command line to print messages when input from float switches was received, and when the pump should have been activated. We eventually added an LED light that would light up when the pump should activate, meaning we no longer had to read the command line output to verify that the pump activation was occurring as it should.

Integration Testing

Our vision for this device is to fully replace a traditional timing circuit with an internet connected, smart alternative. To make the device more user friendly than the original device, we must provide a simple yet functional user interface and activity logging functionality. Because of this structure, we decided to take the "Big Bang Approach" to integration testing. This means that we combined all components and tested them all at once. We tested the triggering of the scripts as cron jobs, the pulling of information from the Open Weather API, and the writing of logs to the device's file system as a single cohesive unit. Everything performed to our standards.

System Testing

For this step, we wanted to test how the system operates in totality. The prototype device is currently in Shane's possession, so he conducted this phase of testing, with input from the team and Mr. McClain on what functions needed to be stress tested. The aspects that were tested include user input to the settings page, requests for log files, proper dispatching of email notifications, proper processing of float switch input, and registration and correct execution of cron jobs. Any immediately notable issues were ironed out.

Acceptance Testing

The culmination of this device's testing phases will be installing our working prototype on Mr. McClain's home septic system. The device will be used 24/7 and be expected to function fully and properly. Once installed, we will be able to monitor its functionality and take note of any shortcomings, bugs, or missing features. If this first stage goes well, we would like to then expand testing beyond the McClains' home, testing with other community members to get more diverse feedback. Once we have gathered this feedback, we will be comfortable launching the device for purchase.

Future Maintenance

Planned Features

Before launching the product on the consumer market, we would like to make a few substantial updates to the system we have developed. Firstly, we will move the storage of user data to AWS S3 buckets. Currently, all user settings and log files are stored locally on each device. We will be adding a user registration process to each device, where a user is guided through the setup of their own personal S3 bucket to contain their log files and settings. This will allow us to keep the data in a more secure location, and will additionally allow users to access their files from afar.

We will also be making use of a second S3 bucket that will manage the codebase for all Aerobicon devices. This bucket will hold the current versions of the source files, and will be used to push new versions to user devices. This bucket will be backed up by yet another bucket, containing old versions of the software. This will allow us to roll back updates if necessary.

The final big change we will make before launch will be migrating to a different standard of hardware. Currently we are using a Raspberry Pi. This type of hardware setup is good for hobby projects, but typically products being sent to market are built on microcontrollers, or proprietary hardware. Before selling this product to customers, we will want to find a more industry-standard hardware solution. We are currently considering the Particle Photon, a microcontroller commonly used for IoT devices. The benefits of switching to this standard would be ease of control, as the Photon would include a companion app allowing users to configure low-level settings such as internet connectivity, as well as reduced costs and complexity. We will also be finalizing the design of our proprietary float sensor system and making it available for purchase. For those users who do not want to purchase our float sensor, we will be developing and offering an adapter that will facilitate the use of existing float switches with our device.

Once the product is launched, we will continue to make changes that improve the user experience. One such feature we have planned is text message notifications. Currently, we offer email notifications only, but once the migration to AWS is complete, we will gain access to the tools provided by that service, including SMS notification support. The user will have the option of opting for one or the other, or both. Another such feature will be allowing for the download of log files. This can be facilitated through both the AWS portal as well as the local Node-based portal.

Security Fixes and Updates

Once we have the AWS integration completed, the users' information will be gated behind a login system and stored on a secure remote server rather than their personal device. This will be a big step forward for the data security of the device. From this point on, any security flaws and exploits will be dealt with as they are discovered. We will be adding a user feedback portal on our website, where users can submit bug reports and participate in public Q&A forums. We will keep a close eye on these reports and respond quickly to security-related bugs, pushing fixes through our main S3 bucket as quickly as possible.

Code Updates

As mentioned, we will be pushing updates to the central codebase via our main S3 bucket. This bucket will contain the current production version of our software. Every day at a set time, the device will check its software package against the version on the S3 bucket. If the device is running an older version, it will download the new version and patch itself. This will require the server process to be rebooted- for this reason, we would want to do it at a time decently far from the scheduled pump activation, likely 6 or more hours after.