

CS Monitoring Project

Requirements Document

https://www.teamliftinc.org

Team Lift

**UP Capstone Team Members**

Pranav Rajan- rajan23@up.edu*,*

Adrian Muth- muth23@up.edu,

Tamsen Dean – deant23@up.ed

Jennifer Brana, brana23@up.edu

Benjamin Chong – chongb24@up.edu

**Client**

Karsten Zuendel

**Faculty Advisor**

Martin Cenek- cenek@up.edu

Table of Contents (functional)

[Introduction 3](#_Toc466295422)

[Product Overview - 3](#_Toc984667493)

[Functional Requirements - 5](#_Toc674304955)

[Non-Functional Requirements - 6](#_Toc1649798595)

[User Interface - 6](#_Toc356920315)

[Preliminary Sprint Schedule – 6](#_Toc690420562)

[High-level Technical Specifications - 7](#_Toc1104541900)

[Budget - 7](#_Toc1152740823)

[Facilities- 7](#_Toc1806557087)

[Ethical Considerations - 7](#_Toc1607554572)

[Conclusion – 8](#_Toc341648904)

[References – 8](#_Toc1812016870)

[We referenced the deliverables and notes left by the civil engineering side of Team Lift. We also consulted with members of the team personally, namely our faculty advisor and client. Dr. Cenek has also done a related project. 8](#_Toc1124177334)

[Other Considerations - 8](#_Toc545235549)

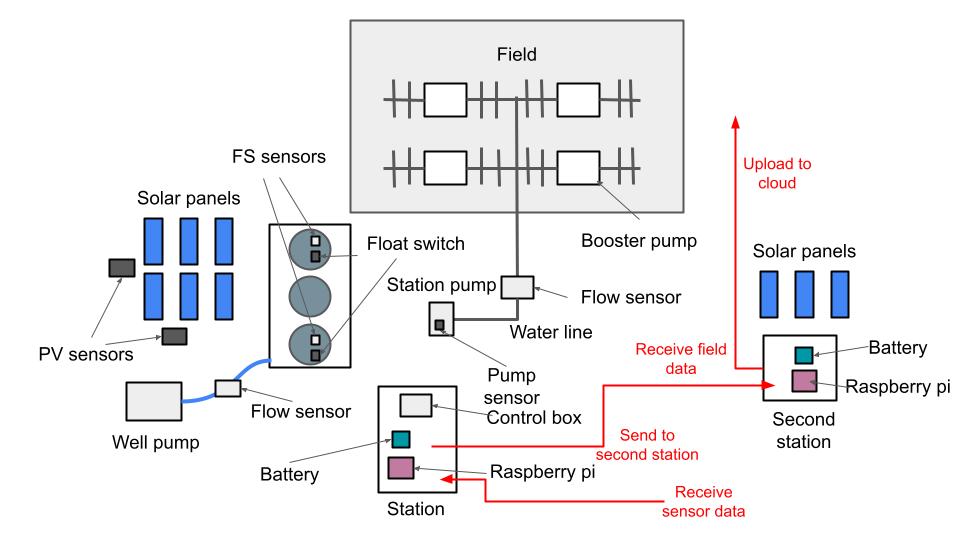
Table of figures:

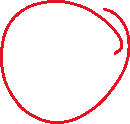
Figure 1 – Product Overview Diagram…………………………………………………………...4

### Introduction

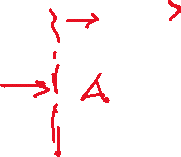
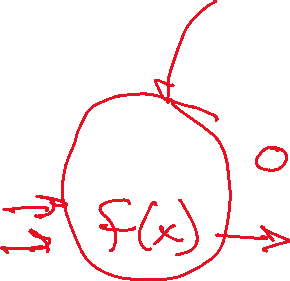
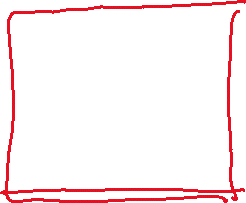
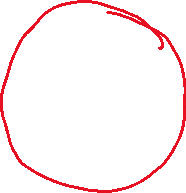
This is a water irrigation monitoring system for a farm in a school zone, in Malawi. We have an array of sensors such as flow sensors, pump sensors, photovoltaic sensors, and float switch sensors, from which we readout data and send to the cloud. This project is a data collection project, and the data is then uploaded to the backend, for review from the users.

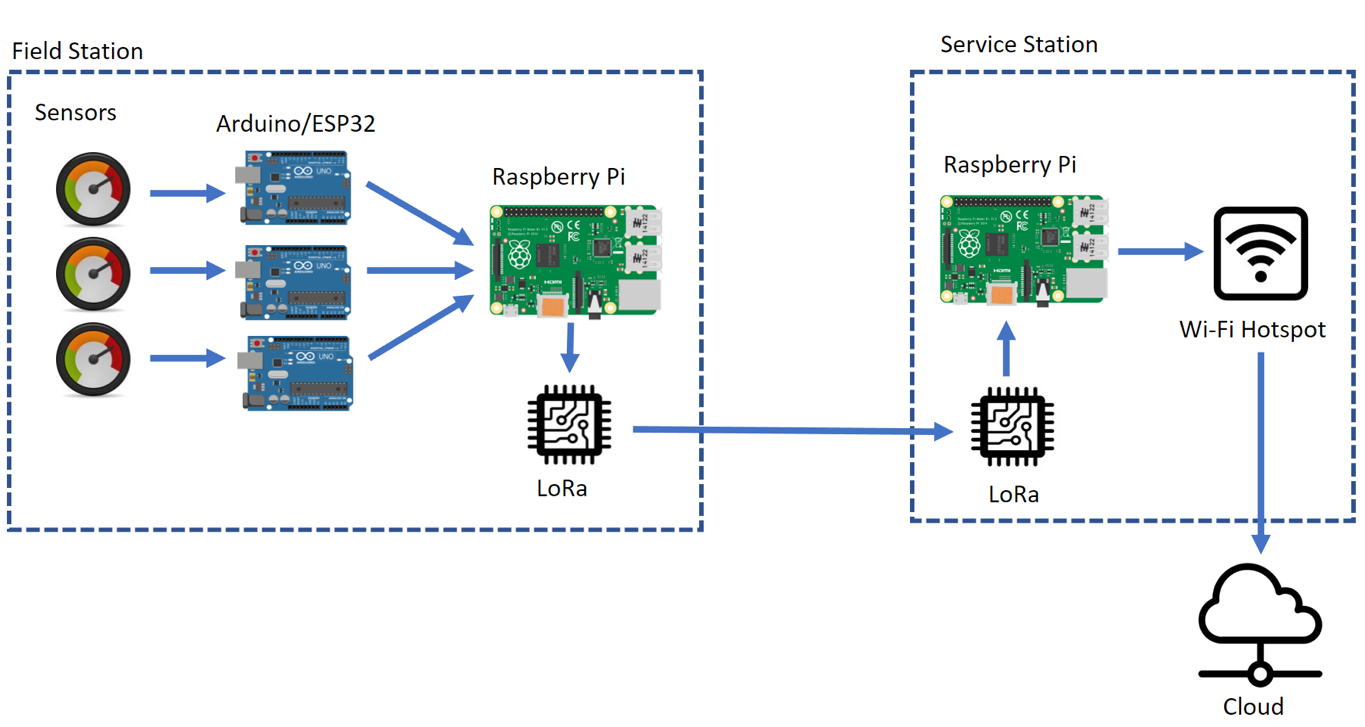
### Product Overview -

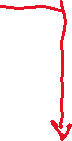
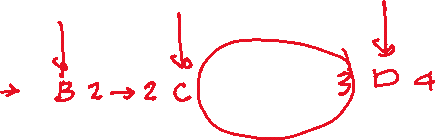
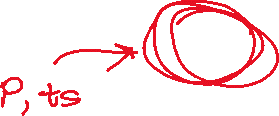
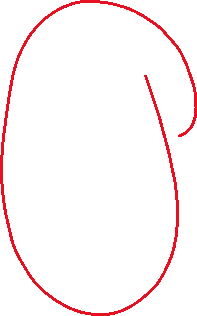




*Figure 1 – Product Overview Diagram*



` ` `



*Figure 2 – Data Flow Diagram*



On the hardware side of things, the Arduino will be connected to all the sensors at the control box at the pump station. The Arduino will periodically read out data from the sensors. These sensors include the flow sensors on the well pump and water line, the pump sensor connected to the station pump, the PV sensor connected to the solar panels, and the Float Switch sensors connected to the float switch on the water tanks. These sensors will all feed into the Arduino. The Arduino will then send all the data periodically to a Raspberry PI via USB connection.

This intermediary raspberry PI will then send the data to another raspberry PI at the second station. This second station has Wi-Fi connectivity for uploading data to the cloud. The intermediary Raspberry PI will communicate to the raspberry PI at the second station, through radio frequency. Once the data arrives to the Raspberry PI at the second station, it uploads this data to the cloud. For the data upload, we will run a python script that will send the data to a google spreadsheets file. This spreadsheets file will show data being updated and accumulated in real time. Another approach we will explore is sending the data to the user through email. These emails will be periodic, likely every 1-2 days.

The user will view the sensor readout data through either accessing the google spreadsheets file, or by viewing the periodic daily emails with the sensor data.

***User Stories -***

* Dirt builds up on the solar panels and the electrical output to the pumps is reduced. Installed sensors detect a voltage drop and this information is uploaded to the cloud. A user checks the cloud and notices the issue and notifies someone to check the panels and they clean off the dirt.
* The filter in the station’s pump needs to be replaced. A sensor interfacing with the pump detects that it is drawing a higher amount of power. Another sensor attached to the pump's output detects that the flow rate has decreased. This information gets pushed to the cloud and is flagged as significant change. A user in Malawi is notified and they check the pump and replace its filter.
* The pump sensor will read a drop in voltage, indicating that the pump is not turned on. However, the flow sensor indicates that water is still flowing. These circumstances indicate that water is leaking from the water pump into the driplines, without the pump being turned on. The user will be alerted of this through an email. On the field they will repair the defective pump that is leaking.
* The pump sensor indicates that the pump is turned on. However, the flow sensors indicate that there is no current of water flowing. These circumstances indicate that there is a blockage in the pump or driplines, because water is not flowing even with the pump being turned on. The user will be notified of this through email. On the field, the blockage inside of the pump or driplines will be investigated and removed.

### Functional Requirements -

* Installed sensors will regularly report data on the irrigation systems state
* Data from sensors will be collected at the station and aggregated
* Data collected at the field station is reliably relayed to a second station that has connection to the internet
* Data received by the second station is uploaded to an accessible database in the cloud
* As a user in Malawi, I need to be able to check the most recent status of the irrigation system as reported by the sensors.
* As a user in Malawi or the US, I need to be able to see the status of the irrigation system as a historical record over time.
* Power is delivered to installed devices at both stations using the existing solar panels
* Power is saved to a battery system allowing for devices to run when solar panels are not outputting electricity



* Base stations:
  + connect a real-time clock connected to the Raspberry Pi’s for a consistent time stamp that does not rely on internet connection.
  + Additional Arduino for each base station as a hardware-based watchdog. Need an independent method (Arduino) to force a reboot in the case that the Pi froze or was shut down for any reason. To prevent a device system crash, Pi has to be reboot every 24 hours. Use built in crontab functionality.
* Data aggregation
  + Raw data received by the pump station from sensor nodes is recorded in text files using Python scripts and timestamped using the RTC time. Following this, the data is sent to the WiFi station.
  + Wifi station aggregates the data using Python scripts which append data into text files. A new daily file is created every day.
* Data sending: need to send data and wait for an acknowledgement before erasing data. If it does not receive an acknowledgement, it must send the data again.

### Non-Functional Requirements -

* The system will be able to be operated and maintained by the local workers.
* The design will be able to withstand the expected weather conditions in the area.
* If a part in the system malfunctions, it can be repaired or replaced.
* In case of a power outage, the system can be rebooted.
* Base station maintenance:
  + Backup raspberry pi SD cards for all Pi’s. Clone the primary card onto backup after system install. In the case of an SD card corruption, the backup card serves as a field replacement.

### User Interface -

We have two potential user interfaces in mind, and we plan on exploring them both. One user interface is having the user go to their email address and have them receive periodic emails every day or every week. These emails will have the sensor readout data in a tabulated format. These sensor readout tables will be in the form of a text file and be compressed into a zip-file to be emailed to the user.

The other user interface is to have the user access a google spreadsheets document. This spreadsheets document will be updated in real time, and have all the most current, up-to-date readout data. There will be only one spreadsheets document, that will show the readout data, rather than multiple, in contrast to periodic emails of sensor data.

### Preliminary Sprint Schedule –

|  |  |  |
| --- | --- | --- |
| **Sprint** | **Sprint Lead** | **Target Goals** |
| Sprint 1  *Aug 27 – Sep 17* | Tamsen Dean | * Communicating with clients and understanding their needs |
| Sprint 2  *Sep 18 – Oct 8* | Adrian Muth | * Set up circuits to communicate data |
| Sprint 3  *Oct 9 - Nov 5* | Benjamin Chong | * Gain access to test flow sensors |
| Sprint 4  *Nov 5 – Dec 3* | Adrian Muth | * Recurrently send flow sensor data to cloud |
| Sprint 5  *Jan 21 – Feb 18* | Jennifer Brana | * Testing and simulating device |
| Sprint 6  *Feb 19 – Mar 18* | Tamsen Dean | * Create first working version of device |
| Sprint 7  *Mar 19 – Apr 8* | Pranav Rajan | * Optimization and reinforcement of system |
| Sprint 8  *Apr 8 – Apr 29* | Benjamin Chong | * Final fixes and project sendoff |

### High-level Technical Specifications -

Data is expected to be collected every 15 to 60 minutes.

Data is expected to be sent to the cloud every 24 hours.

Data is kept in the cloud at least up to the last 2 years.

### Budget -

Estimated costs associated with our project:

|  |  |
| --- | --- |
| Products/services | Cost |
| Freenove Ultimate Starter Kit for ESP32-WROVER x 4 | $215 |
| Arduino Uno x 8 | $280 |
| Raspberry Pi 4 x 4 | $620 |
| MicroSD x 8 | $160 |
| ESP 32 Feather Board x 4 | $110 |
| Cables | $100 |
| Adafruit Liquid Flow Meter x 4 | $80 |
| LoRa 32 (<https://heltec.org/project/wifi-lora-32-v3/>) x 4 | $120 + $20 for shipping |
| Power Bank x 2 | $200 |
| 12V Lead Acid Battery | $20 |
| Step Down 12V to 5V Buck Converter | $15 |
| Pelican Box x 2 | $100 |
| **Total Estimated Price** | $2135 |

It’s worth noting that some of these products have already been provided at no cost by the university.

### Facilities-

This information is based on the civil engineering team’s final design report. This is what they have already implemented that we will be reading data from.

* Grundfos submersible solar powered groundwater pump (primary)
  + Pumps water into tanks
* Grundfos CRFlex solar pumping system (secondary)
  + Adds pressure and draws water from tanks
* Solar panels
* Pump charger
* Polyethylene plastic 10,000L water storage tanks
* Drip and manual irrigation networks
* Operation and Maintenance manual

### Ethical Considerations -

Our project is directly related to improving the lives at Saint Mary’s in Malawi. As such, we have an ethical obligation to deliver a viable project. However, our project is unlikely to be abused in an immoral manner. An ethical consideration is to make our project safe with the wildlife in Malawi. For example, we should consider possibilities such as animals chewing through electrical wires and harming themselves.

### Conclusion –

This document is a snapshot created 5 weeks in of what we aim to accomplish as a final product. We have spoken with our clients and made decisions based on their needs and our capabilities.

### References –

## We referenced the deliverables and notes left by the civil engineering side of Team Lift. We also consulted with members of the team personally, namely our faculty advisor and client. Dr. Cenek has also done a related project.

## Other Considerations -

Though we currently plan to deliver all the listed requirements above, there are stages that require reaching out to companies for parts—to test and simulate the product—that have yet to receive a response. In cases such as these, the acquisition of materials is out of our control.