The Official Group - Project Proposal

Home AC Monitoring and Reporting System

Project Owners

Eric Easley
Cole Fellers
Matthew Fredericksen
Ian Ocasio
Luis Roman

Under the direction of

Shraddha Piparia
Obianuju Okafor

University of North Texas
Department of Computer Science and Engineering

Date

September 14, 2020

Project Description

Purpose

To expose health and usage information of residential air-conditions systems to predict maintenance and failure.

Description

Filter change prediction

Using a manifold air pressure (MAP) sensor between the filter and the blower, we can determine how dirty the filter is, and project over time when it will need to be replaced. We can send email alerts (perhaps with filter spec information, and even a purchase link).

Temperature differential measurements

By measuring the difference between the outgoing and return air temperatures, we can determine if the A/C unit is cooling effectively. The ideal temperature differential is between 16 and 22 degrees. This temperature difference is the "evaporator Delta T". If the delta T is too high, it indicates there is an air-flow restriction (dirty filter, clogged evaporator core, ducting problem). If delta T is too low, it is likely low refrigerant, leaky ducting, leaking reverse valve or compressor issues. Regardless of the root cause, tracking the shift of delta-T over time will provide a means of detecting an issue well ahead of the unit failing, perhaps even allowing one to roughly predict when failure will occur. An example would be if there is a slow refrigerant leak; when the unit is first used after winter, the statistics will expose a pending issue even though the unit appears to be functioning well in terms of cooling the house.

Run-times and energy usage

In a multi-zone house, it is possible that children may adjust A/C set-points to unreasonable settings, causing long run-times even though the A/C unit is functioning properly. Tracking run-times and reporting anomalies. Optionally, we can monitor the call signals to the zone controller as opposed to just the unit call signal. This will identify which zones are calling at abnormal frequencies or durations. We will need to capture call signals at the zone controller, as different zones have different flow characteristics and will perturb our delta-t measurement. We will need to determine through observation how these interact.

Optional: Smart t-stat integration

By integrating with REST APIs provided by smart thermostat makers such as Nest and Honeywell, one can monitor set-points and schedules, and alert when settings are set to unreasonable values.

Structure

A Raspberry Pi will be used with external sensors to collect data from the AC system. This data will be exposed through a REST API using Flask (Python). The Raspberry Pi will not host any other processes.

A Prometheus time series database (TSDB) will pull data from this API at regular intervals. Prometheus has a built-in AlertManager which will be used to send notifications when certain trends or data points are detected.

A web app will be created using React JS. The purpose of the web application will be primarily to view useful statistics over an arbitrary range (perhaps disjoint, e.g. last 3 summers). It should also allow configuring alert trigger thresholds. Useful statistics include: average daily runtime, count of on-off cycles, average temperature differential, temperature differential variance, average air pressure, weather data, and more. The web app will provide a configuration interface for managing who receives notifications.

The API, TSDB, and React app are all distinct processes that may be developed simultaneously. The TSDB and React app may be hosted on the same or different computers for development purposes.

Depending on what is learned about database querying with React, another Flask API may be created to act as a middleman between the web app and the TSDB. Preferably, this API would be hosted on the same machine as the TSDB.

Project Plan

Facilities

The most important space for our project will be a controlled environment for proper testing on our detection hardware/software to get consistent and eventually accurate readings before the detection system is ready to be used in an uncontrolled environment. Any other space would be our own homes as meetings are strictly online and a common meeting area is not necessary.

Staff

The effort of every group member may vary depending on a task as some portions of code may be easier to implement than others, but we require 100% completion from all members for the project to be completed in a timely manner. If a group member doesn't have a portion of code by the time all of the code is being implemented, the system will fall apart until that portion is completed, so completion is a huge part of the staff effort, the rest includes participation, sharing ideas, and cooperation.

Hardware

A Raspberry Pi will be used to read **pressure and temperature sensors**. These items total at about $60, and have already been purchased by our project sponsor.

Personally owned computers will be used for development and deployment of the application.

Software

React is the software that most users will see as it is the front end. React provides a powerful framework for interactive web applications that will facilitate the implementation of our code. Open-source React widgets, such as React-Date-Picker, may be used as the need arises.

Flask is a lightweight web framework that will be used for creating the necessary APIs, such as for reading data from the Raspberry PI and possibly querying the TSDB.

Prometheus fits into our code nicely as it is a systems monitoring and alerting toolkit with a built in alert manager providing data for our systems as well as information for users that doesn't include the normal GUI.

All software used during development will be open-source (i.e. free), so the only costs involved are for time, as outlined in the following Gantt chart.

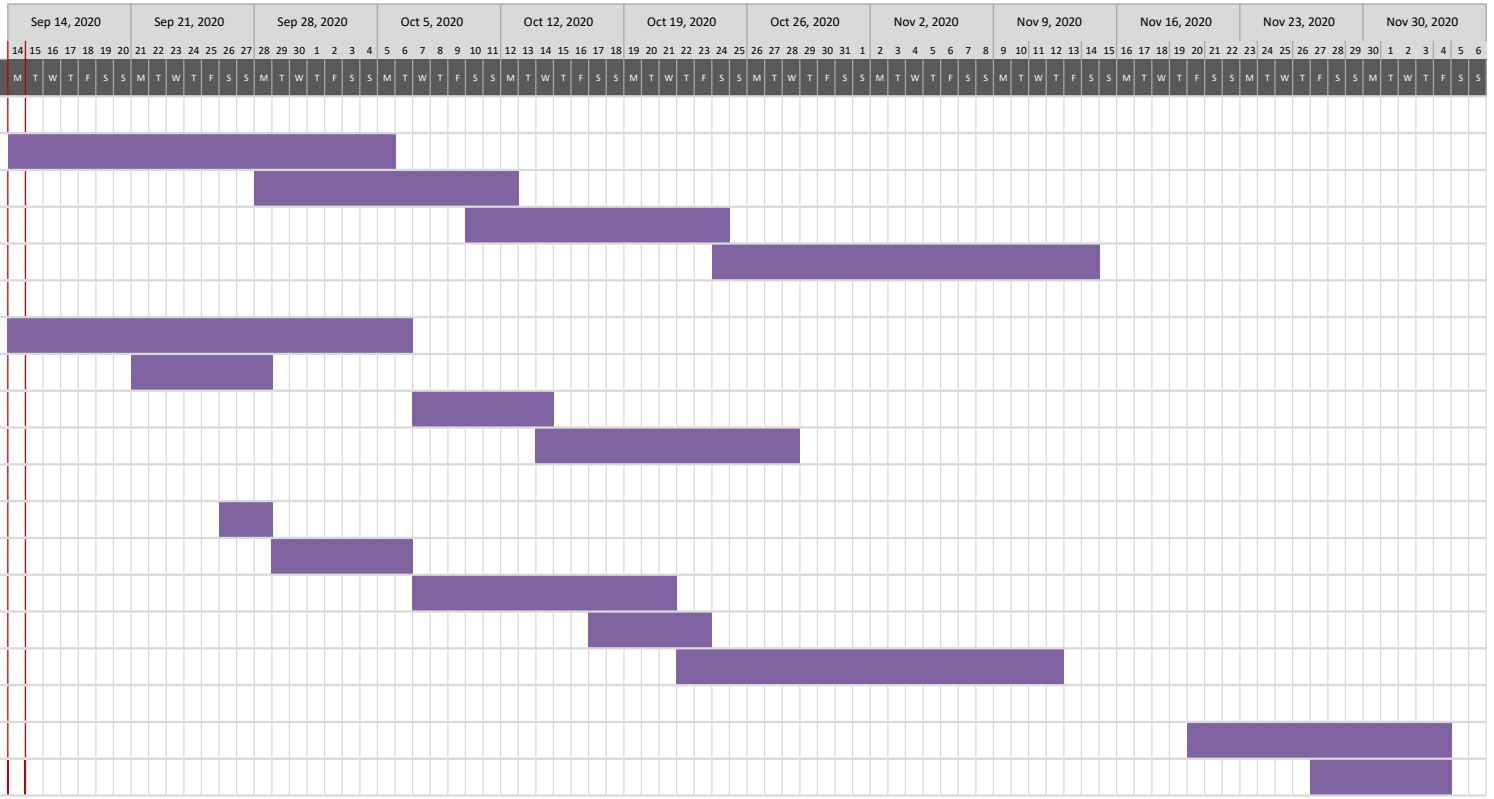
Home AC Monitoring & Reporting System

The Official Group
Project Timeline

Project Start: Mon, 9/14/2020

Display Week: 1

TASK	START	END
Front end		
Learn React	9/14/20	10/5/20
Create base web app to build on	9/28/20	10/12/20
Create live view of sensor data	10/10/20	10/24/20
Create historical view (i.e. statistics)	10/24/20	11/14/20
Sensors/API		
Research sensor information	9/14/20	10/6/20
Create Flask API (mock response data)	9/21/20	9/28/20
Add real sensors to API	10/7/20	10/14/20
Create middleman API for DB queries	10/14/20	10/28/20
TSDB		
Install	9/26/20	9/28/20
Configure sensor-API pulling	9/29/20	10/6/20
Create long-range test data for front end testing	10/7/20	10/21/20
Write database queries	10/17/20	10/23/20
Configure AlertManager	10/22/20	11/12/20
Project Report		
Write report	11/20/20	12/4/20
Create slides	11/27/20	12/4/20



Risk Management

Risk	Mitigation Strategy
Our team lacks familiarity with React.	● Have videos for team members to watch so that they are familiar with React. ● The first activity in the project will be to learn React.
Differences in schedules may cause difficulties with meeting as a group.	● Be flexible with whoever is able to be there, we may not need every person for every meeting. ● Meeting notes can be shared with any students unable to attend. ● Extra class time could be used as a backup meeting time to ensure everyone could attend.
Sensors could cause damage to the AC unit.	● Make sure that any and all objects inside the AC unit are tightly secured. ● Add redundant measures to ensure the sensor remains secure.
The system may send a false alert in extreme weather.	● Figure out how airflow would be effected in these conditions and factor that into the responsiveness of the sensor.
Damage to the sensor could cause incorrect data and need for a replacement.	● Ensure the sensor is installed in a location that makes replacement a relatively easy process. ● If the sensor is reporting inconsistent data, it could send an alert to the web interface.

Lack of familiarity with React.

The instance could arise where a team member's lack of experience with the React environment could limit not only their productivity, but their ability to complete the task required of them. In this situation, the other members of the group would be able to point said member in the right direction to tutorials online to help them familiarize themselves with the material.

Differences in schedules may cause difficulties with meeting as a group.

The key issue at the heart of this risk is the fact that we are all students with many classes that can often conflict. This can often make scheduling meetings that can be attended by every member an issue. However, depending on how we divide up the work, certain issues that we might meet about might not require all members. For example, if we choose to have only 2 students working on the web interface, and an issue arises related to the design of the interface, the other 3 students may not need to attend. If this is the case, meeting notes can be taken by

the members that meet and distributed to the members that were not involved. We also have whatever time that would be available after class to meet if need be.

Sensors could cause damage to the air conditioning unit.

The most potentially unpredictable risk involved is how the sensors are going to remain secure once inside the AC unit. If they come loose and are propelled by the unit's fan, they could cause damage to either the filter or the unit itself. To prevent this from taking place, we would have to make sure we have a sturdy method of securing the sensors to the unit. Redundant methods of securing the devices would also go a long way to preventing this from occurring.

The system may send a false alert in extreme temperature cases.

Extreme weather conditions could put an especially high strain on an AC unit, for example, it could be 120 degrees outside and the AC cools the house down to 70 to 80 degrees. In these extraordinary conditions, the fault detection techniques based on our airflow analysis could be rendered ineffective. To mitigate this, we would need to figure out exactly how the airflow is effected, so we can either factor that into our fault detection system, or send a disclaimer that results could be inconclusive.

Damage to the sensor could cause incorrect data and need for a replacement.

One risk associated with any project that involves hardware is the integrity of the hardware. Before installing the hardware into the unit, we would have to determine the safest place for the hardware, as well as the point where the hardware would be easiest to replace. The data that is being sent back could also be analyzed for unnatural responses, or some pattern that could indicate a faulty sensor. The system could also run periodic checks to look for any kind of failures.

Team Roles

All team members will act as Developers. After considering team roles, we decided that this project was not complex enough to require more distinct divisions of labor, especially since a core goal of our team is for each of us to gain experience with all of the technologies we will be using.

In lieu of having a Project Manager, each team member has agreed to use GitHub Projects throughout the duration of the project to facilitate issues/progress tracking. This will allow us to quickly notice if any section of the project is lagging and requires assistance.

Regular usage of Git will allow us to work on distinct project requirements simultaneously, and allow each team member to contribute to various tasks as necessary.