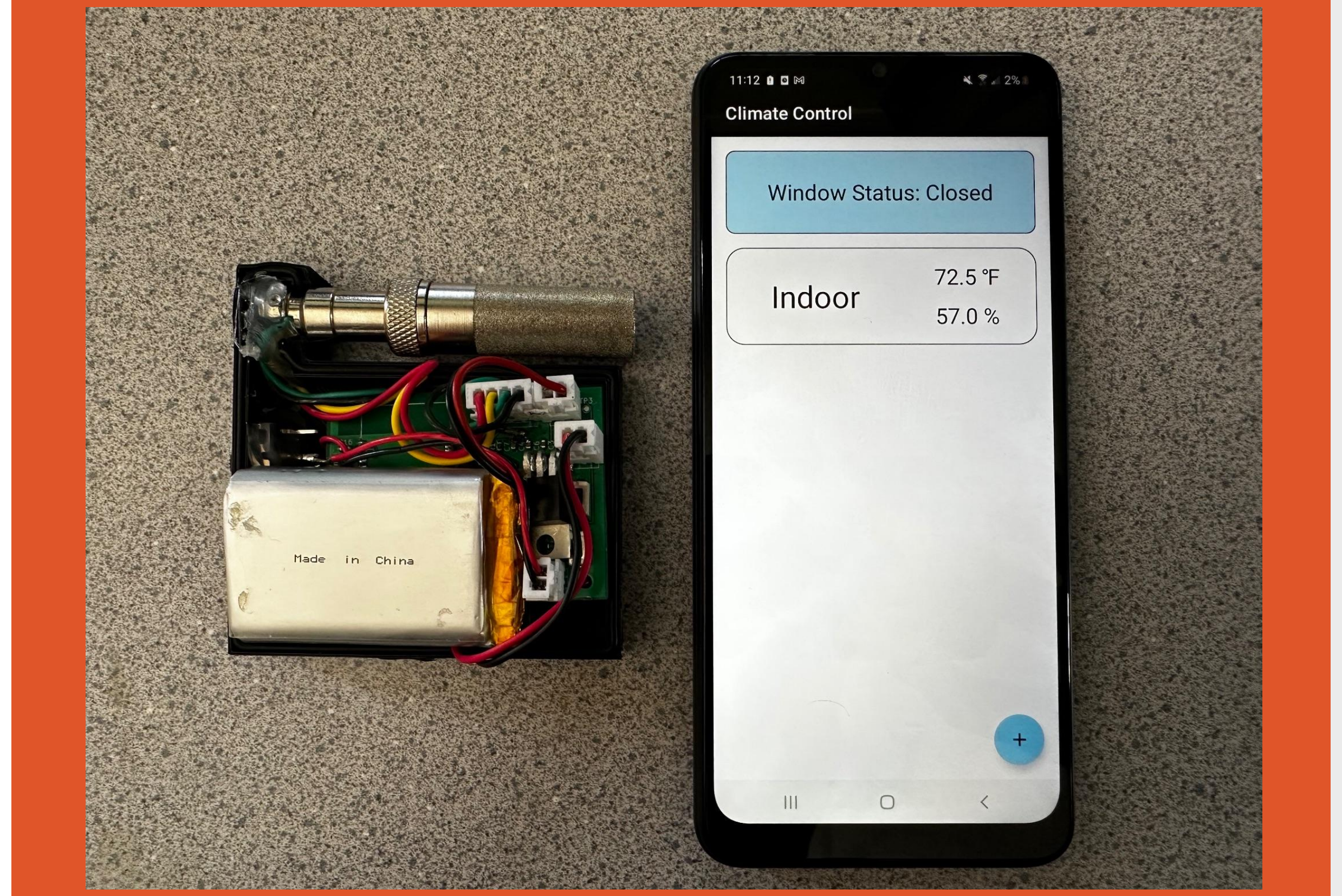
Sensor System for In-Home Climate Control

Modular IOT system that provides users with window opening and closing recommendations to control climate



System

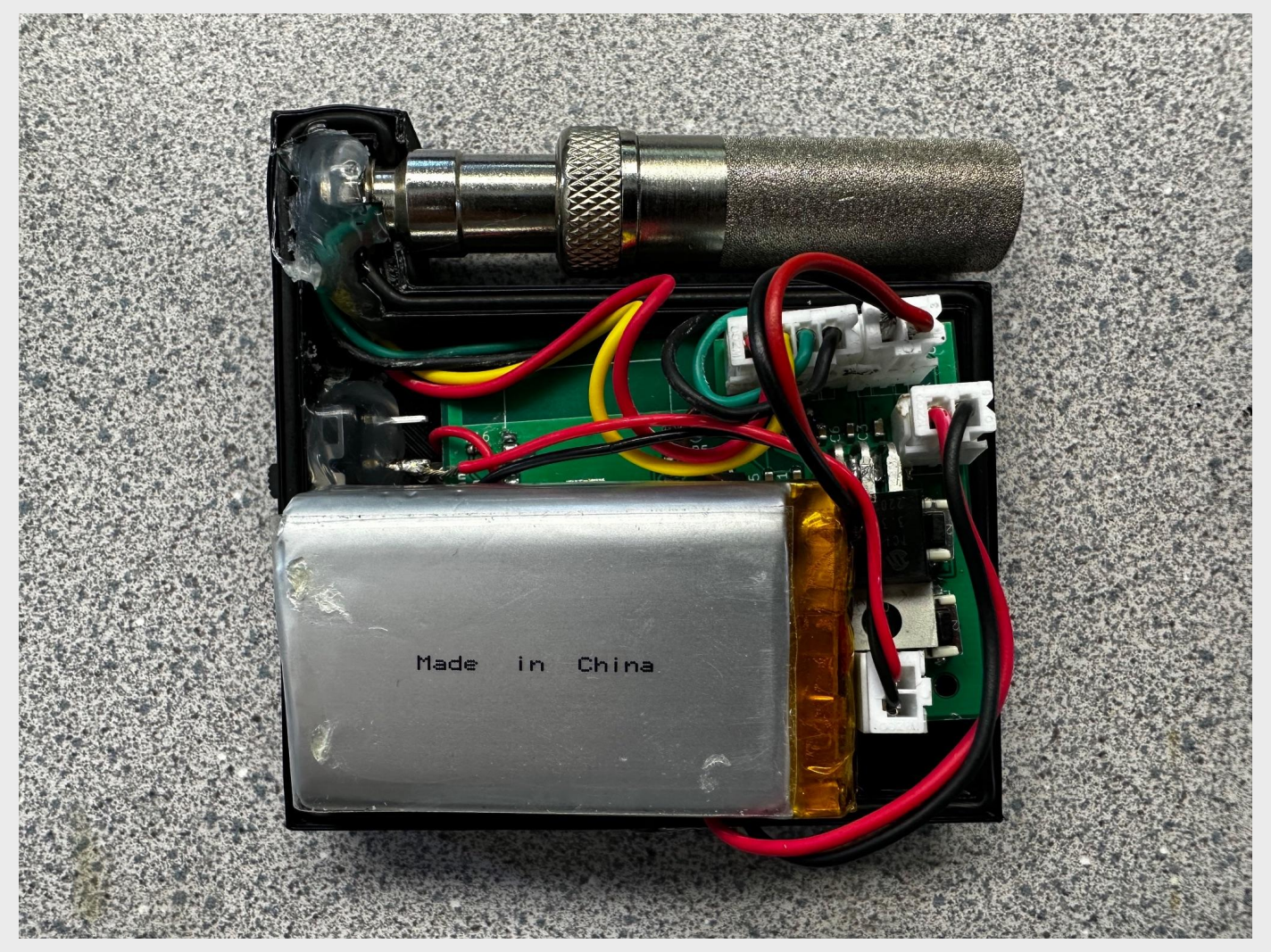
Engineering Requirements

- **Battery Life:** The system will operate for at least one week on a single charge.
- **Device Labeling:** The application will have the ability to differentiate between nodes.
- **Device Size:** The sensor subsystem will be no larger than 3x3x1 inches.
- **GUI:** The application will display accurate temperature and humidity data in a way that is easy to read.
- **Notifications:** The application will notify users of a change in the window status in way that is easy to understand.
- **Sensors:** The system will collect temperature and humidity data within 5° Celsius and 10% humidity of actual.
- **Weather Proof Enclosure:** The system will have IPx4 weather proofing.
- **Wireless:** The sensor subsystem will pair with the application wirelessly.

Oregon State University

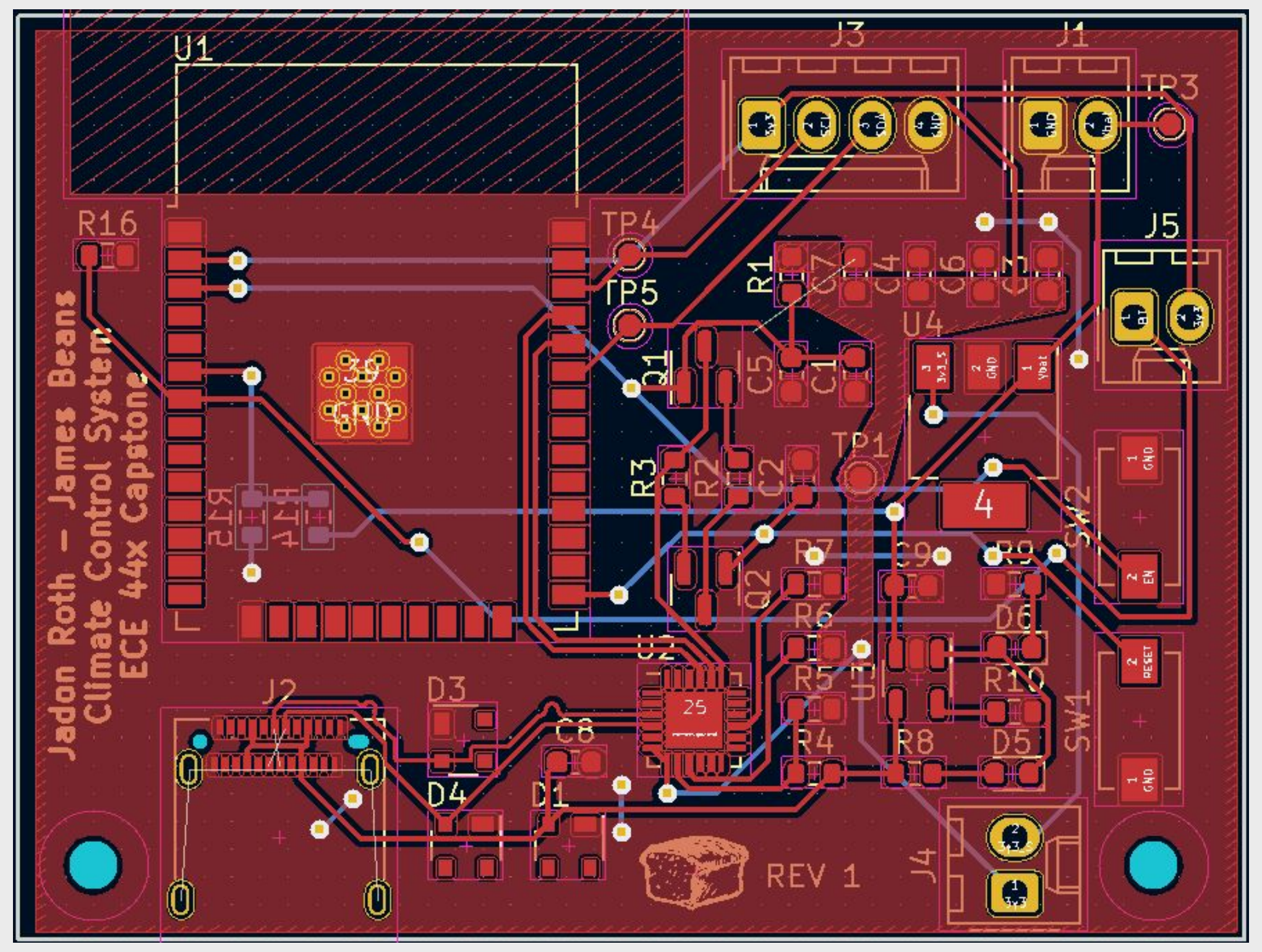
Project Summary

As the cost of electricity and natural gas increases, the effective use of windows is important to regulate interior temperature and save users money. Our system is designed to just that! Through the use of several indoor and outdoor nodes collecting temperature and humidity data, we can notify users when to naturally cool and heat their homes!



Sensor

How it Works



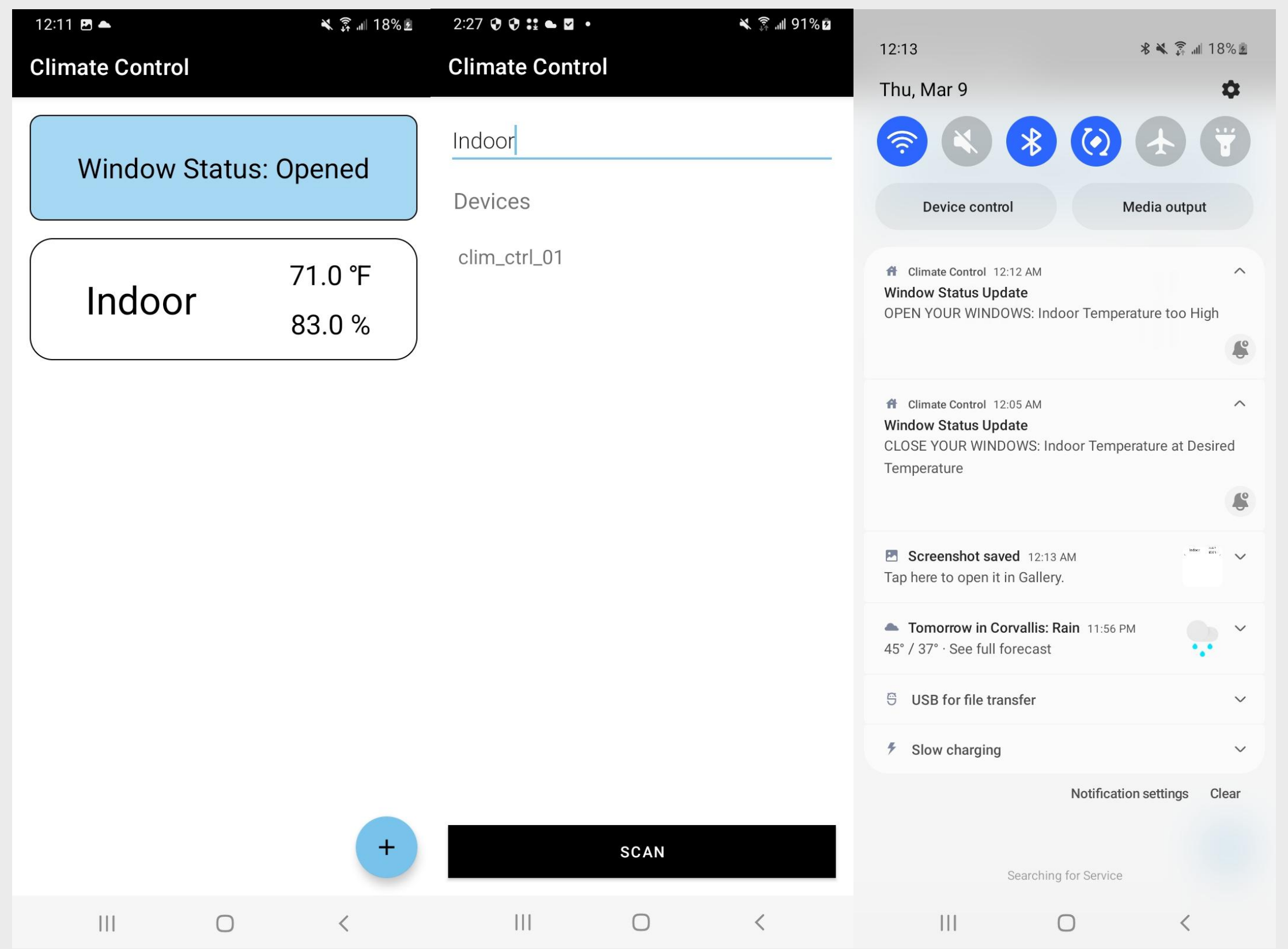
PCB

Sensor Subsystem (Hardware)

- Utilizing the SHT30 sensor, we can collect temperature and humidity data with an accuracy of ±0.5°C.
- The ESP32 was the designated microcontroller for this system, capable of both wifi and bluetooth connectivity.
- With an 2000 mAh battery, this device is capable of 8 days of use without charge. Rechargeable via USB C!
- Up to 30 days of offline storage using ESP32!
- Packaged in a IPx4 waterproof enclosure capable of keeping our electronics safe!

Application Subsystem (Software)

- Built for Android OS
- Displays current window status and node temperature and humidity data.
- Allows for adding nodes via bluetooth with custom naming schemes.
- Each sensor contains an info page where data points stored in the cloud are queried to create line graphs.
- Data is updated live! As data is received, it is plotted and updated app side.
- Users are sent a push notification on their phone of any changes in the window status.



12:11 18%

Climate Control

Window Status: Opened

Indoor 71.0 ℉ 83.0 %

2:27 91%

Climate Control

Indoor

Devices

clim_ctrl_01

SCAN

12:13 18%

Thu, Mar 9

Device control | Media output

Climate Control 12:12 AM
Window Status Update
OPEN YOUR WINDOWS: Indoor Temperature too High

Climate Control 12:05 AM
Window Status Update
CLOSE YOUR WINDOWS: Indoor Temperature at Desired Temperature

Screenshot saved 12:13 AM
Tap here to open it in Gallery.

Tomorrow in Corvallis: Rain 11:56 PM
45° / 37° · See full forecast

USB for file transfer

Slow charging

Notification settings Clear

Searching for Service

Application

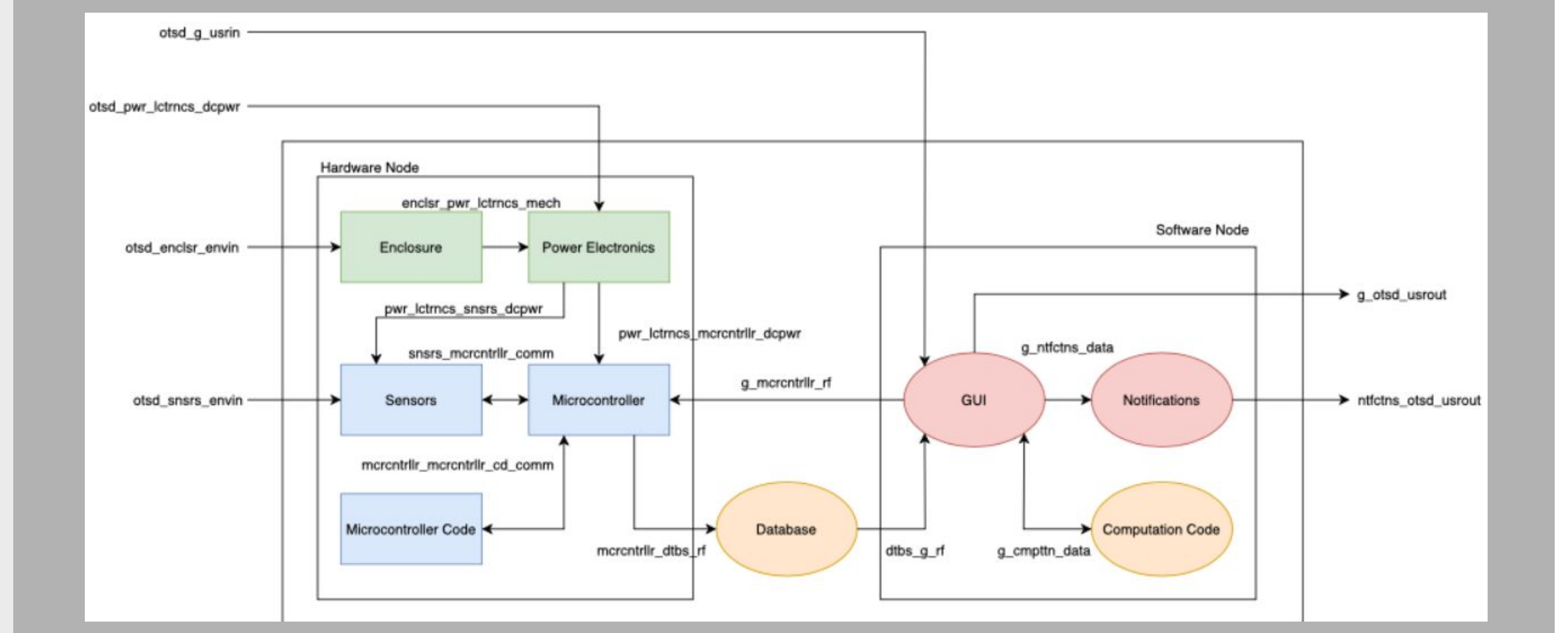
Meet Team 20

Blake Wiker (Top Left):
wikerb@oregonstate.edu
Blake worked on database and app side computation and integrating the two. He also implemented background threads to complete these tasks.

Yousif Albaker (Top Right):
albakery@oregonstate.edu
Yousif was responsible for implementing the GUI and notification parts of the application. This involved creating easy to read and understand visual elements and application navigation.

James Beans (Bottom Left):
beansja@oregonstate.edu
James designed all the power electronics ensuring efficient and compact energy conversion. He also built our 3D printed enclosure ensuring a waterproof system!

Jadon Roth (Bottom Right):
dirksjad@oregonstate.edu
Jadon oversaw everything microcontroller. He designed the onboard ESP32 and related circuitry and created the onboard sensor logic. He also boiled down the final circuits into a single PCB.



otsd_g_usrin
otsd_pwr_lctrncs_dcpwr
Hardware Node
enclsr_pwr_lctrncs_mech
otsd_enclsr_envin
Enclosure
Power Electronics
pwr_lctrncs_snsrs_dcpwr
pwr_lctrncs_mcrcntrllr_dcpwr
snsrs_mcrcntrllr_comm
otsd_snsrs_envin
Sensors
Microcontroller
g_mcrcntrllr_rf
mcrcntrllr_mcrcntrllr_cd_comm
Microcontroller Code
mcrcntrllr_dtbs_rf
Database
dtbs_g_rf
Software Node
g_otsd_usrout
g_ntfctns_data
GUI
Notifications
ntfctns_otsd_usrout
g_cmpttn_data
Computation Code