Second Prototype Test Report

Team 27
Domus: Smart Home Control

Venkata Sai Sreeram Andra vssa8989@bu.edu
Anirudh Singh ansingh@bu.edu
Vansh Bhatia vansh@bu.edu
Akhil Bongu akbongu@bu.edu

Table of Contents

1 Required Materials

1.1 Hardware:

- Raspberry Pi 5 (with 32GB Amazon Basics microSDXC Class 10 memory card)
- 2 ESP32 Devkit S1 Micontrollers
- Raspberry Pi 4B
- BME688 Sensor
- Raspberry Pi AI Camera IMX 500
- HC-SR501 Motion sensor
- TP Link KP115 Smart Plug
- Verizon Wifi Router

1.2 Software:

- Python scripts:
    - Collecting real-time sensor data (hosted using a Flask Server - using websocket for real time alerting)
    - Kasa API to link smart plug data to our personal app
    - Streamed live video footage using Gstreamer Pipeline
    - Face Recognition using dlib + OpenCV
    - Object Detection using MobileNet SSD V2
    - Activity Recognition using MoveNet + LSTM
- React Native:
    - Integrated sensor data, face recognition functionality through iOS mobile application.

2 Set Up

The Raspberry Pi 5 serves as the central microprocessor, coordinating data collection and processing for the smart home system. All sensors are connected to ESP32 edge microprocessors powered directly through the wall outlets to ensure circuit safety. The ESP32 devices wirelessly transmit sensor data to the Raspberry Pi 5, where Python scripts process the information. The Raspberry Pi AI camera is connected to a Raspberry Pi 4 through the MIPI CSI camera port, which serves as its dedicated edge processor, and streams live footage wirelessly with the central Raspberry Pi 5. The data collected from all sensors is displayed in real-time on a React Native application, which provides interfaces for theft detection, motion sensing, environmental metrics and energy tracking. The theft detection system utilizes a face recognition model built with OpenCV and the face_recognition library from the dlib library.

2.1 Pre-testing Setup Procedure

1. Power On the Raspberry Pi 5 and Raspberry Pi 4
    a. Connect the Raspberry Pi 5 to a power source using its USB-C cable. Ensure the preloaded microSDXC card is already inserted.
    b. Connect the Raspberry Pi 4 to a power source using its USB-C cable.
    c. Wait for the Raspberry Pi 5 to boot up and verify its connection to the local Wi-Fi network.

2. Sensor Validation
    a. Ensure that the BME688 (air quality sensor) and HCS5-501 (motion sensor) are getting powered from the wall outlets and wirelessly connected to the Raspberry Pi 5 via their respective ESP32 edge processors.
    b. Run the pre-installed Python scripts on the Raspberry Pi 5 to confirm that the sensor data is being transmitted and received correctly. Check the console logs for errors or missing data.
    c. If using the Kasa smart plug, make sure the appliance powered by it is well connected and the plug's blue light turns on.
3. React Native App Setup
    a. Launch the React Native app on the connected mobile device or emulator. Ensure the app loads to the home screen.
4. Feature Verification
    a. Navigate to each screen in the app to verify that:
        i. **Motion Sensing**: The app reflects motion detected by the HC-SR501 sensor.
        ii. **Theft Detection**: Specific screen within the mobile application displays real time alerts of unknown/authorized personnel appearing in the camera's field of view.
        iii. **Energy Tracking**: The screen is set up to display current power/total energy consumption metrics in the form of time graphs once data integration is complete.

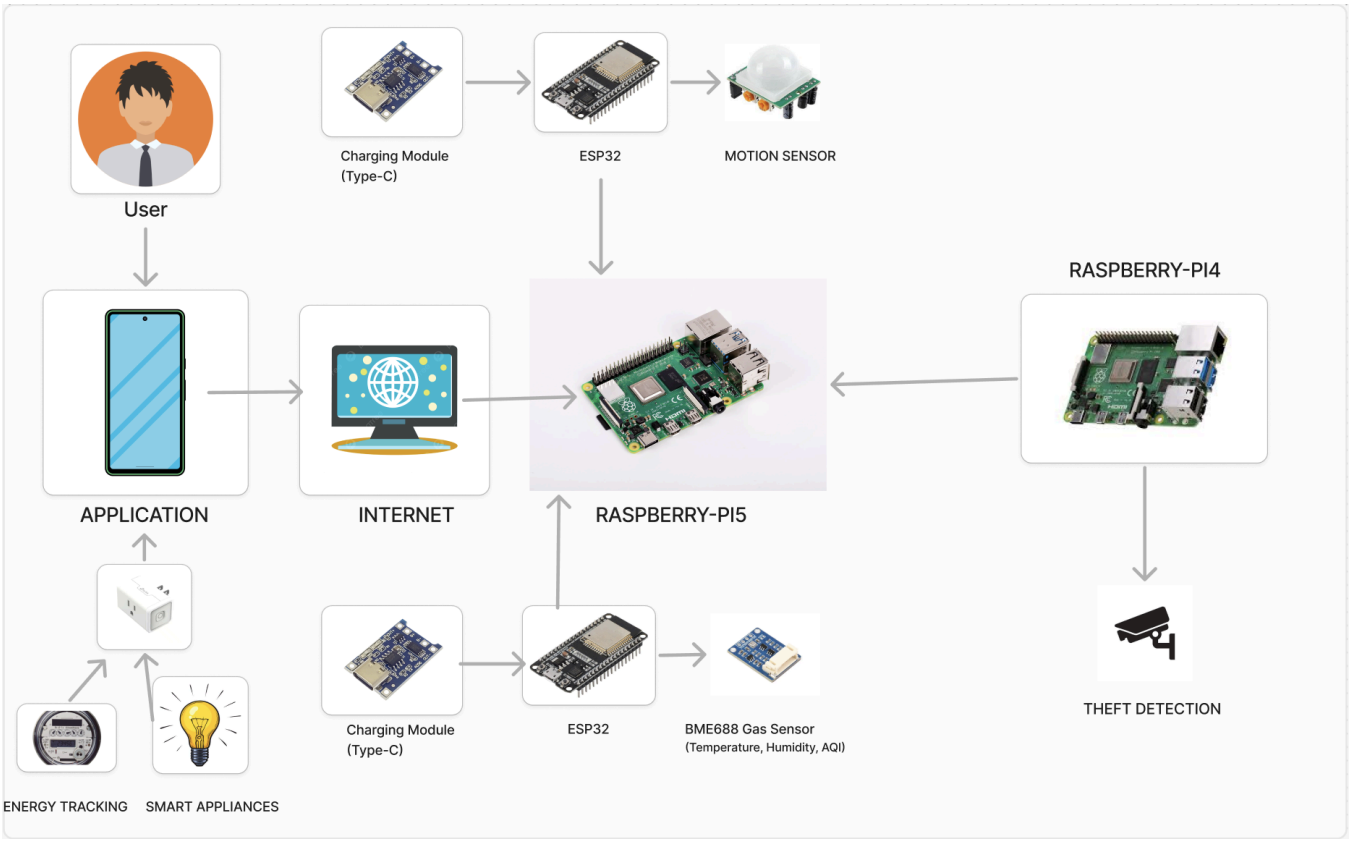


Figure 1: *Illustration of Setup and Process Flow*

2.2 Testing Procedure

1. **Live Motion Detection:** Wave hand in front of sensor (from up to 7 metres away)
2. **Face Recognition:** Should generate alerts for unknown/authorized personnel present in the camera's field of view in real time.
3. **BME688 Sensor:**
    - **Temperature/Humidity Measurement:** Should display temperature (in ℃) and humidity (in %)
    - **Air Quality:** Gas resistance (in Ohms) & Air Quality indicator meter
    - **Atmospheric pressure** (in hPa)

4. **Energy tracking:** Should generate real time power/energy consumption graphs for any load connected to the smart plug.

3 Measurable Criteria

1. **Motion Sensor:** Detected any movement in front of the broad range of the sensor and updated the status of the sensor on the app simultaneously.

2. **Theft Detection Camera:** The face detection system successfully distinguished between authorized and unauthorized person and a real-time update for the same was on the app.

3. **Temperature and other indoor environment data:**
    - Displayed temperature in Celsius and and humidity in percentage
    - AQI meter with six different zones for indicating the air quality in a given room or area
    - Displayed atmospheric pressure in hPa

4. **Energy Tracking:** The energy screen on the app should show energy consumption and overall power usage graphs and current values of a particular outlet and the connected device and has ability to turn the connection “ON” or “OFF” through the mobile app.

4 Conclusion

4.1 Evaluation

The prototype's functionality was as expected. The motion sensor and BME688 both relayed real time values to the React-Native iOS application. The camera module was able to distinguish between an authorized and unauthorized person, and was able to display an alert to the mobile application in real-time with minimal latency. The smart plug also provided real time energy consumption data to the mobile application, and allowed for remote on/off capabilities.

All the sensors connected to the Raspberry Pi 5 via the edge microcontrollers seamlessly as expected. The response from the sensors was fast and without hitches, as shown in the figures below.

4.2 Problems

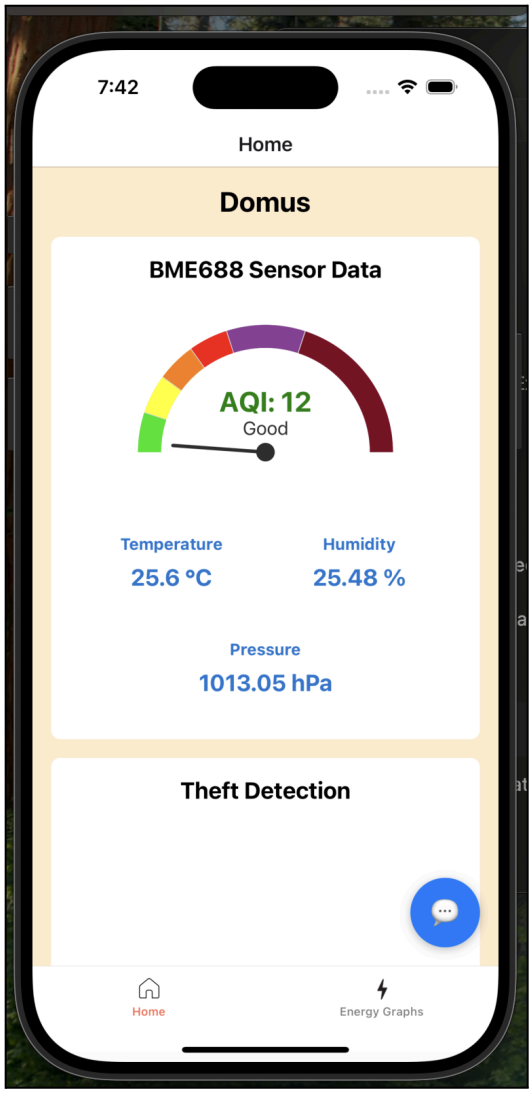
- **Network Connectivity Issues**
    - Challenge: Difficulty in maintaining stable communication through a single network.
    - Improvement: Implement a personal router with LAN connection to integrate seamlessly with BU Wi-Fi.
- **Ergonomic and Durable Housing**
    - Challenge: Designing compact and user-friendly housing units.
    - Improvement: Optimize design using durable materials and modular components to enhance usability and longevity.

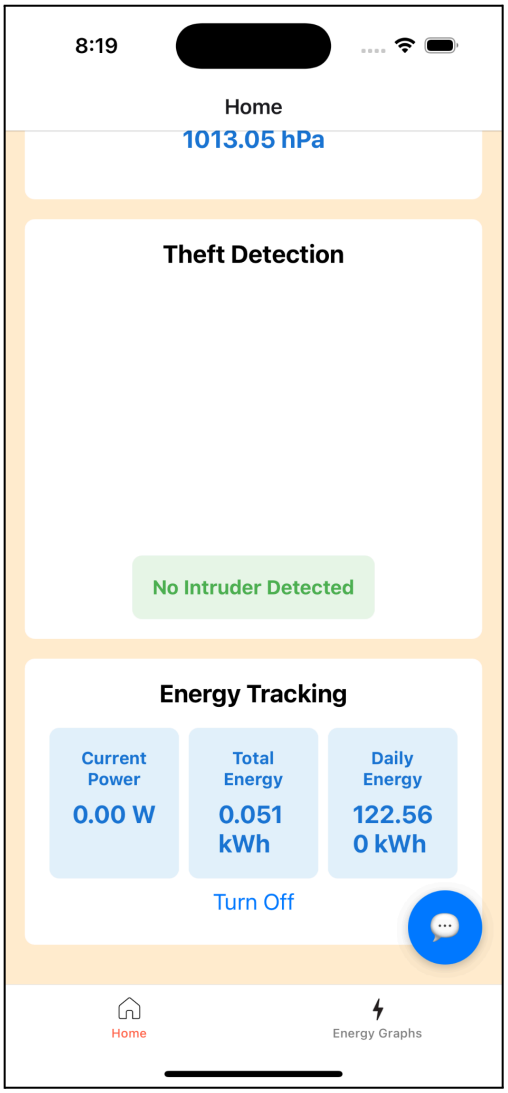
- **User Interface Design**
    - Challenge: Developing an intuitive and interactive UI.
    - Improvement: Conduct further iterative development to improve user experience and functionality.
- **Camera Streaming Quality**
    - Challenge: Ensuring high-quality, low-latency streaming from the camera.
    - Improvement: Experiment with alternative streaming protocols, stream directly to the app, and execute inference on the camera itself to reduce latency.

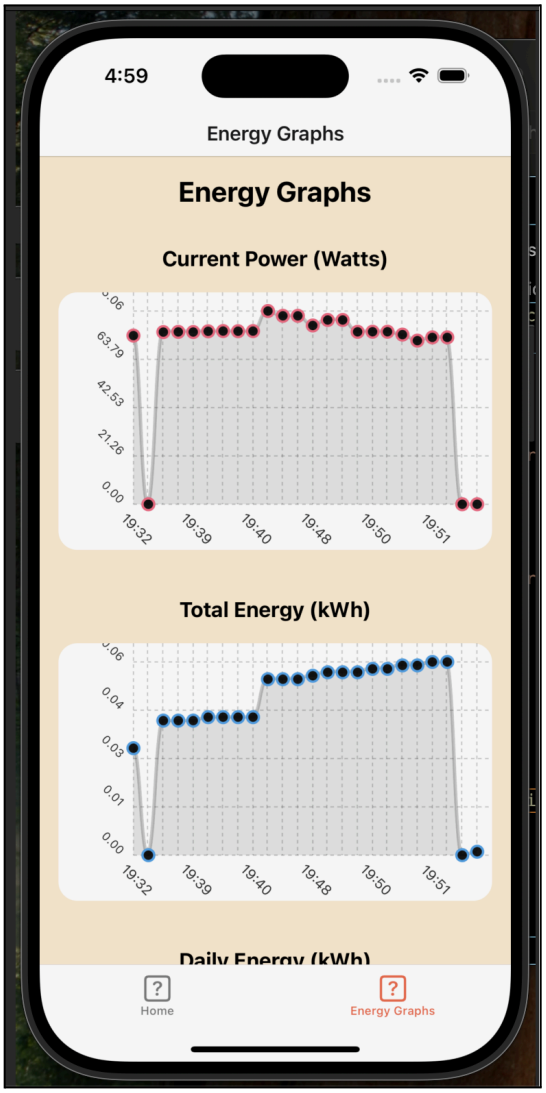
4.3 Next Steps

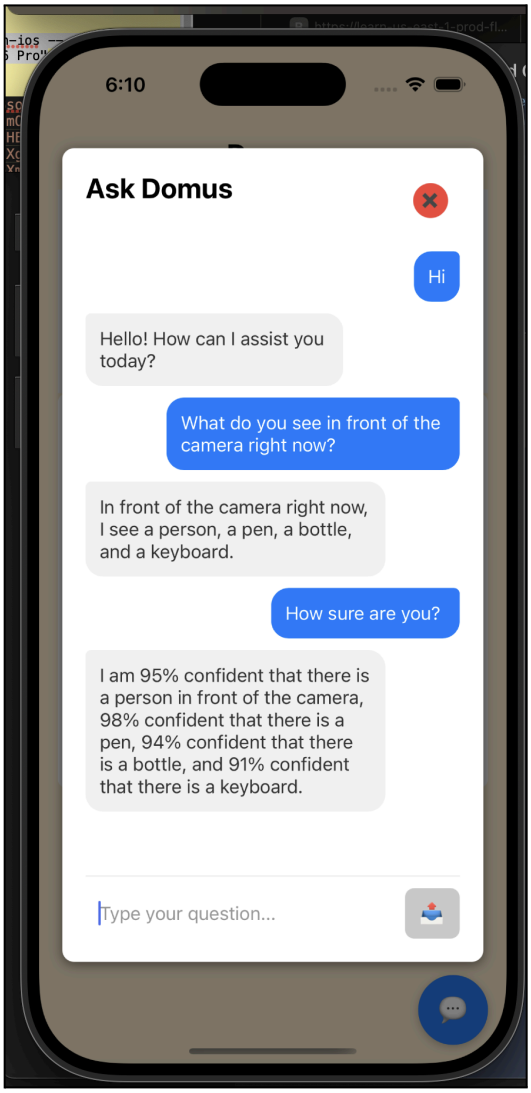
- Optimize the in app LLM and integrate voice commands
- Improve the UI and make it more user friendly
- Train a more refined computer vision learning model for higher accuracy in object detection and further use to efficient theft detection system
- Develop/3D-print a new plug-in housing unit for our edge sensors and a holder for the camera
- Apply a linear regression model to do predictive analysis with energy consumption
- Figure out how to use the BU WiFi network on our local wifi router via LAN cable to simultaneously use the LLM and camera footage on the application

5 Appendix


7:42
Home
Domus
BME688 Sensor Data
AQI: 12
Good
Temperature
25.6 °C
Humidity
25.48 %
Pressure
1013.05 hPa
Theft Detection
Home
Energy Graphs


8:19
Home
1013.05 hPa
Theft Detection
No Intruder Detected
Energy Tracking
Current Power
0.00 W
Total Energy
0.051 kWh
Daily Energy
122.56 0 kWh
Turn Off
Home
Energy Graphs


4:59
Energy Graphs
Energy Graphs
Current Power (Watts)
63.79
42.53
21.26
0.00
19:32
19:39
19:40
19:48
19:50
19:51
Total Energy (kWh)
0.04
0.03
0.01
0.00
Home
Energy Graphs


6:10
Ask Domus
Hi
Hello! How can I assist you today?
What do you see in front of the camera right now?
In front of the camera right now, I see a person, a pen, a bottle, and a keyboard.
How sure are you?
I am 95% confident that there is a person in front of the camera, 98% confident that there is a pen, 94% confident that there is a bottle, and 91% confident that there is a keyboard.
Type your question...