Mobile App Design Final Project

RESIDENTIAL IOT SUITE (RIS)

A Modern Approach to
Remote Security and
Enviromental Monitoring

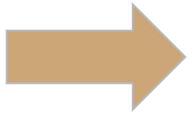
0610101
張天碩 Chang, Tien-Shuo

OUTLINE

- INSPIRATION
- OVERVIEW
- DEVICES & METHOD

INSPIRATION

- Housing appliances with IOT functionalities are costly in both hardware and software, with low compatibility across different brands
- What if we, as students, want to apply some basic IOT functionalities to our dormitory and upgrade our living experience?



OVERVIEW

1. Security System

2. Environmental Monitoring (Temperature, Humidity)

3. User Monitor & Control Interface

1. Security System

- **Human Detection**
    - Triggers when human detected at entry point (ex: doorway, window)

→ Sound alarm and send snapshot remotely back to user-end

→ Owner can deactivate when returning and activate when leaving

2. Environmental Minitoring

- **Temperature**
  - Measures environment temperature in °C

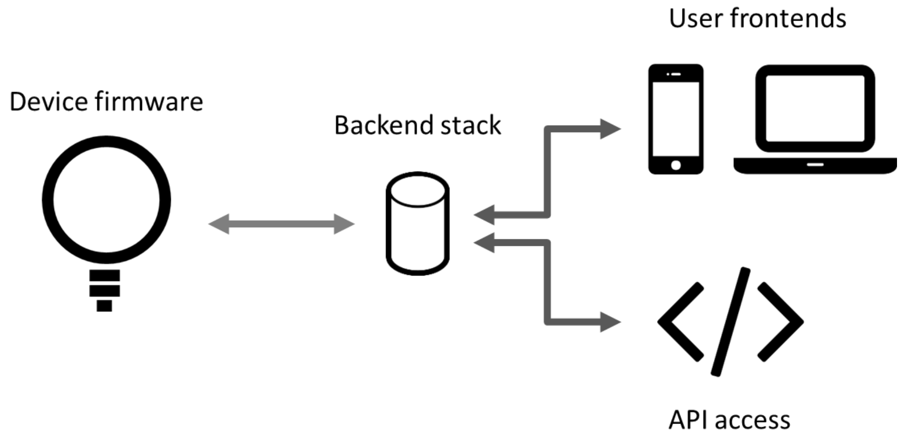
- **Humidity**
  - Measures environment temperature by %

→ Dashboard design
→ Dynamically refreshes to give most recent update

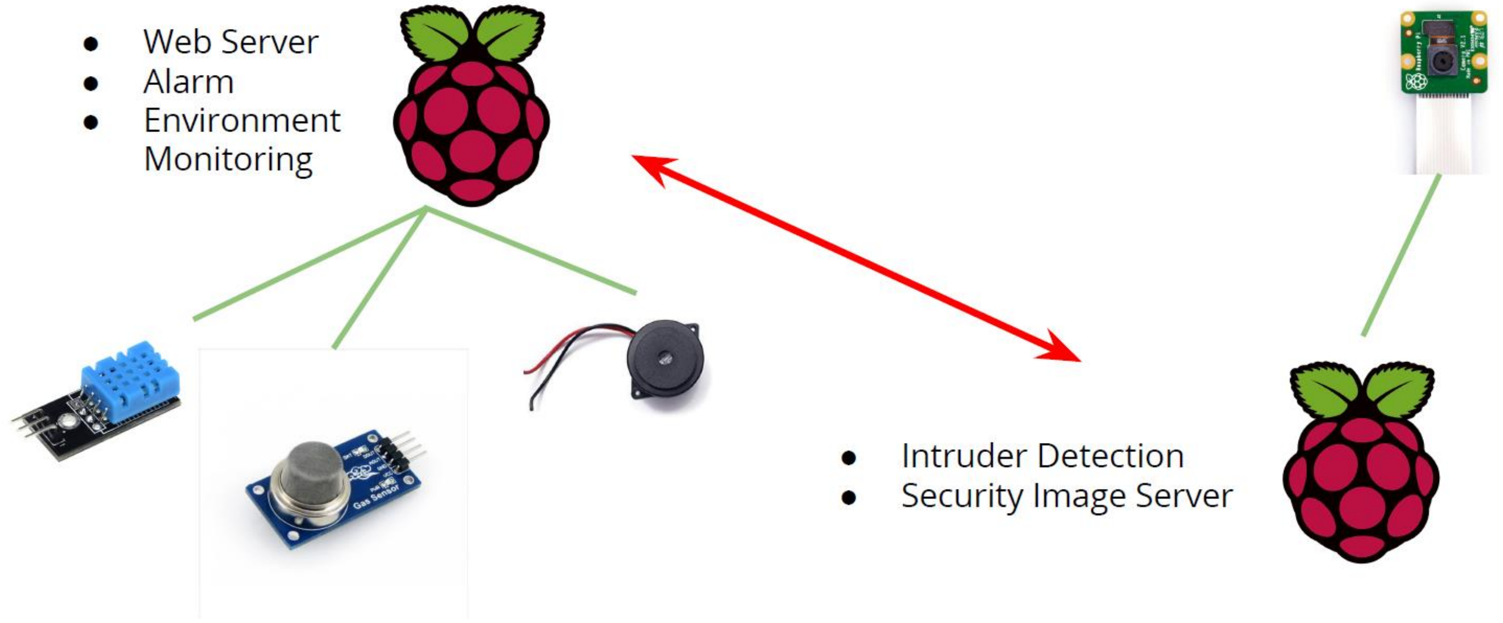
3. User Monitor & Control Interface

Web application with frontend to display data and backend to interact and retrieve information from devices.

- React, Python, MySQL



DEVICES & METHOD



- Web Server
- Alarm
- Environment Monitoring

- Intruder Detection
- Security Image Server

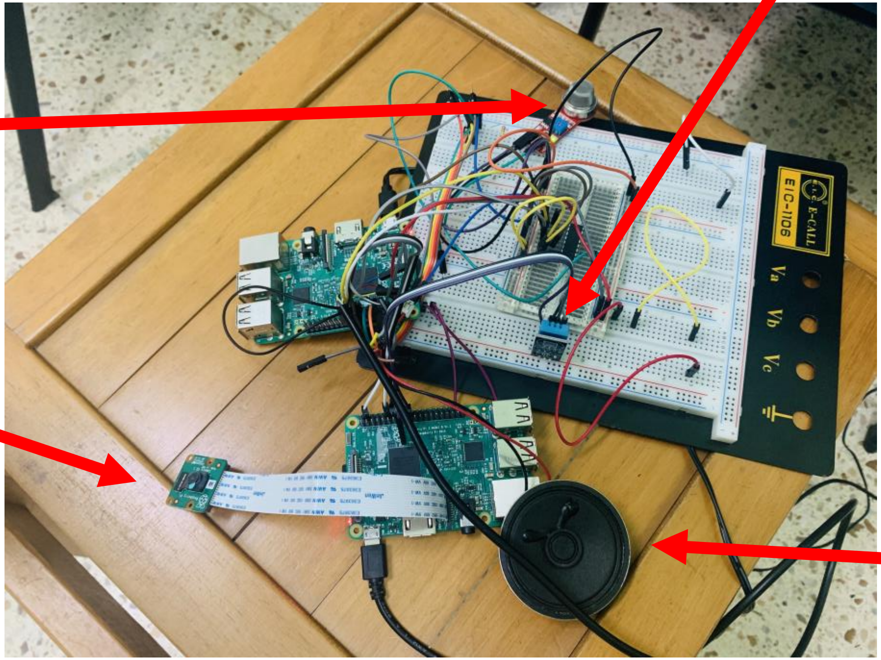
Device Setup

DHT11 Temperature and Humidity Sensor

Air Quality Sensor (Wasn't stable so scrapped it in the final production ☹)

Pi Camera for taking security shots

Buzzer to act as alarm

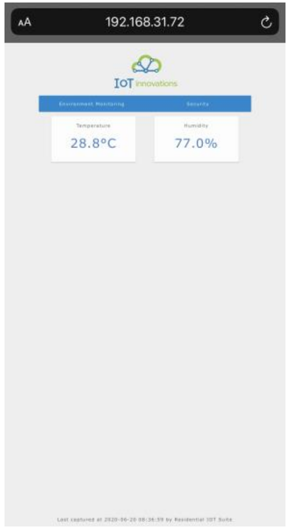


FRONTEND - USER INTERFACE

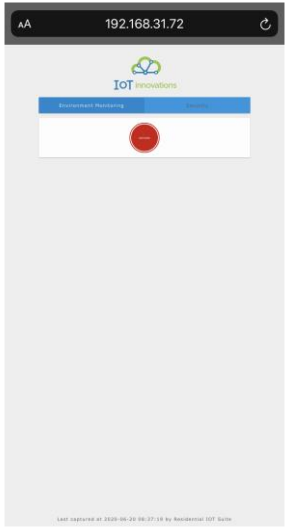
- Request data from Python Flask backend
- Mobile interface friendly

React JS

Mobile Web Version

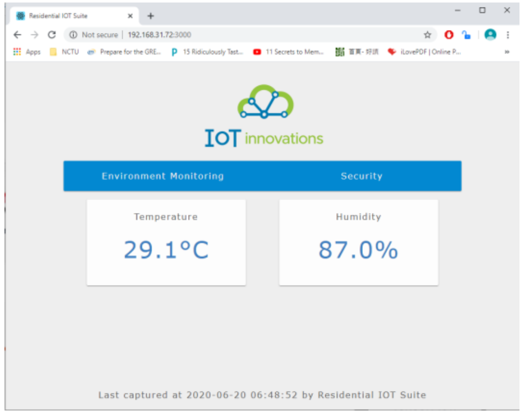


Last captured at 2020-06-20 08:36:59 by Residential IOT Suite

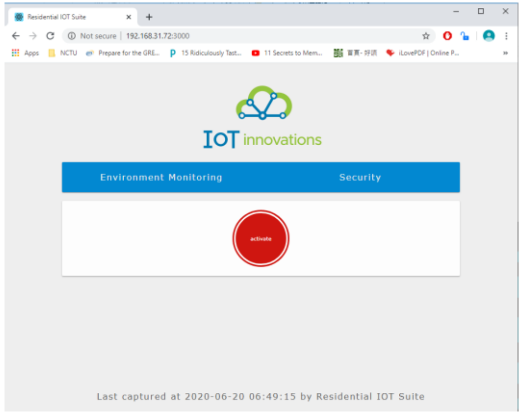


Last captured at 2020-06-20 08:37:19 by Residential IOT Suite

Desktop Web Version



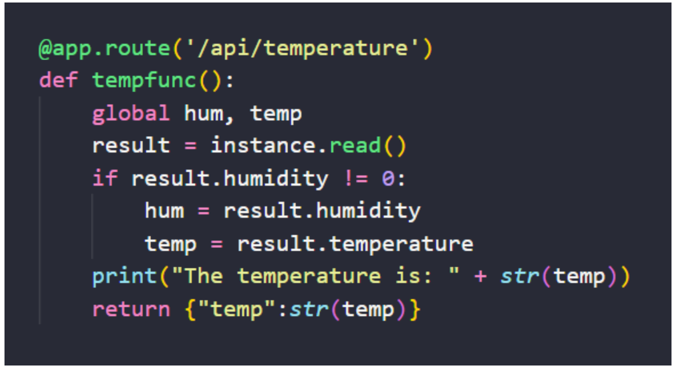
Last captured at 2020-06-20 06:48:52 by Residential IOT Suite



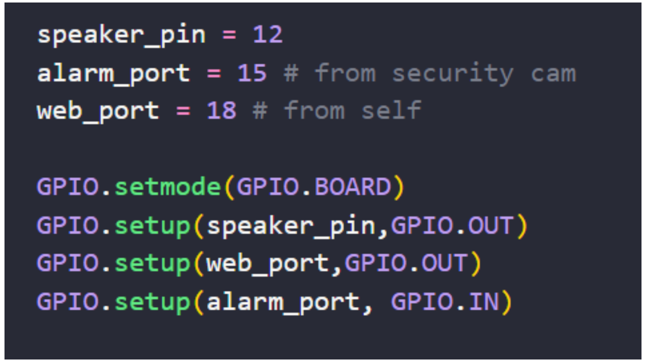
Last captured at 2020-06-20 06:49:15 by Residential IOT Suite

BACKEND - PYTHON FLASK

- Handle Data
- Control Devices
- Serve as API to UI Interface (React frontend)



```python
@app.route('/api/temperature')
def tempfunc():
    global hum, temp
    result = instance.read()
    if result.humidity != 0:
        hum = result.humidity
        temp = result.temperature
    print("The temperature is: " + str(temp))
    return {"temp":str(temp)}
```



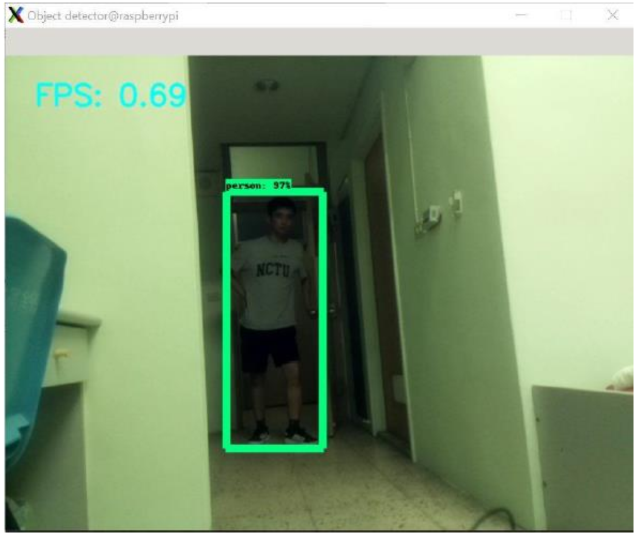
```python
speaker_pin = 12
alarm_port = 15 # from security cam
web_port = 18 # from self

GPIO.setmode(GPIO.BOARD)
GPIO.setup(speaker_pin,GPIO.OUT)
GPIO.setup(web_port,GPIO.OUT)
GPIO.setup(alarm_port, GPIO.IN)
```

HUMAN DETECTOR

- The alarm will be triggered upon detecting human presence (when the detected possibility is greater than a designated threshold) in activated state
- Use TensorFlow Object Detection API (pretrained on COCO dataset)
- Modify the code to do Human Detection

TensorFlow



Demonstration Video

* **Video link :** https://drive.google.com/file/d/1lYFVfktvXHIgrJRTpz9NxvQEu6vo5bkX/view?usp=sharing