



Lacey Pacey: Tracking Foot Health and Preventing Injuries

Thien Vo
MEng, Design & Technology Innovation

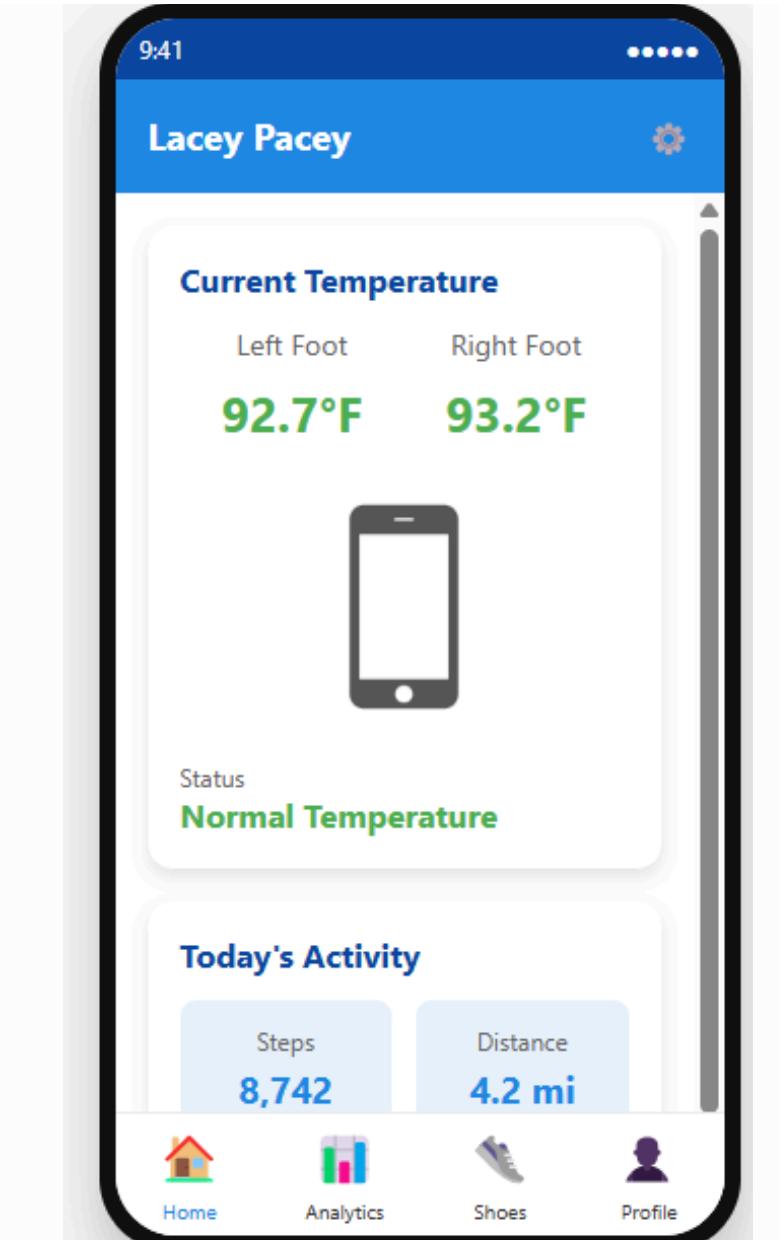
Every day, millions ignore subtle foot pain until it becomes a serious injury.

In sports, 15% of injuries affect the foot alone.

Journal of Athletic Training



MIND THE GAP



Introducing Lacey Pacey

A smart, sensor-powered system that continuously monitors your foot health—so you can prevent injuries before they happen

Smart Features for Smart Athletes



Temperature Monitoring

Continuously monitors foot temperature to detect early warning signs of overheating and potential injury



Real-time Alerts

Receive immediate notifications when your feet reach dangerous temperature thresholds



Performance Analytics

Track your foot health metrics over time to optimize your training and prevent recurring issues

Key Metrics Tracked



Temperature

Measures the warmth of your feet, which can indicate inflammation or poor circulation.



Pressure

Monitors the distribution of weight across your feet, detecting areas of high pressure that may lead to blisters or calluses.



Steps

Tracks your daily activity levels, providing insights into your overall foot health and mobility.

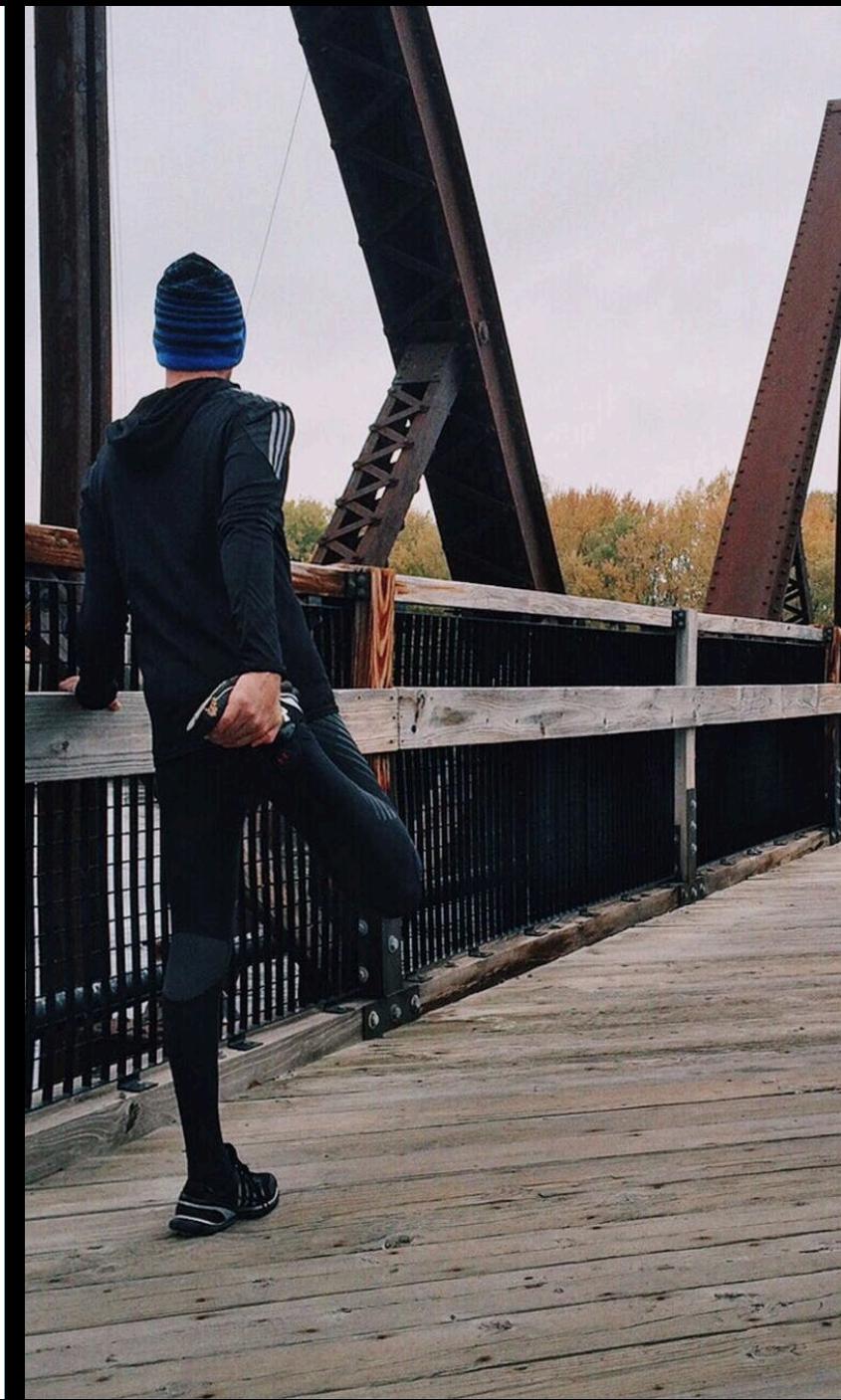
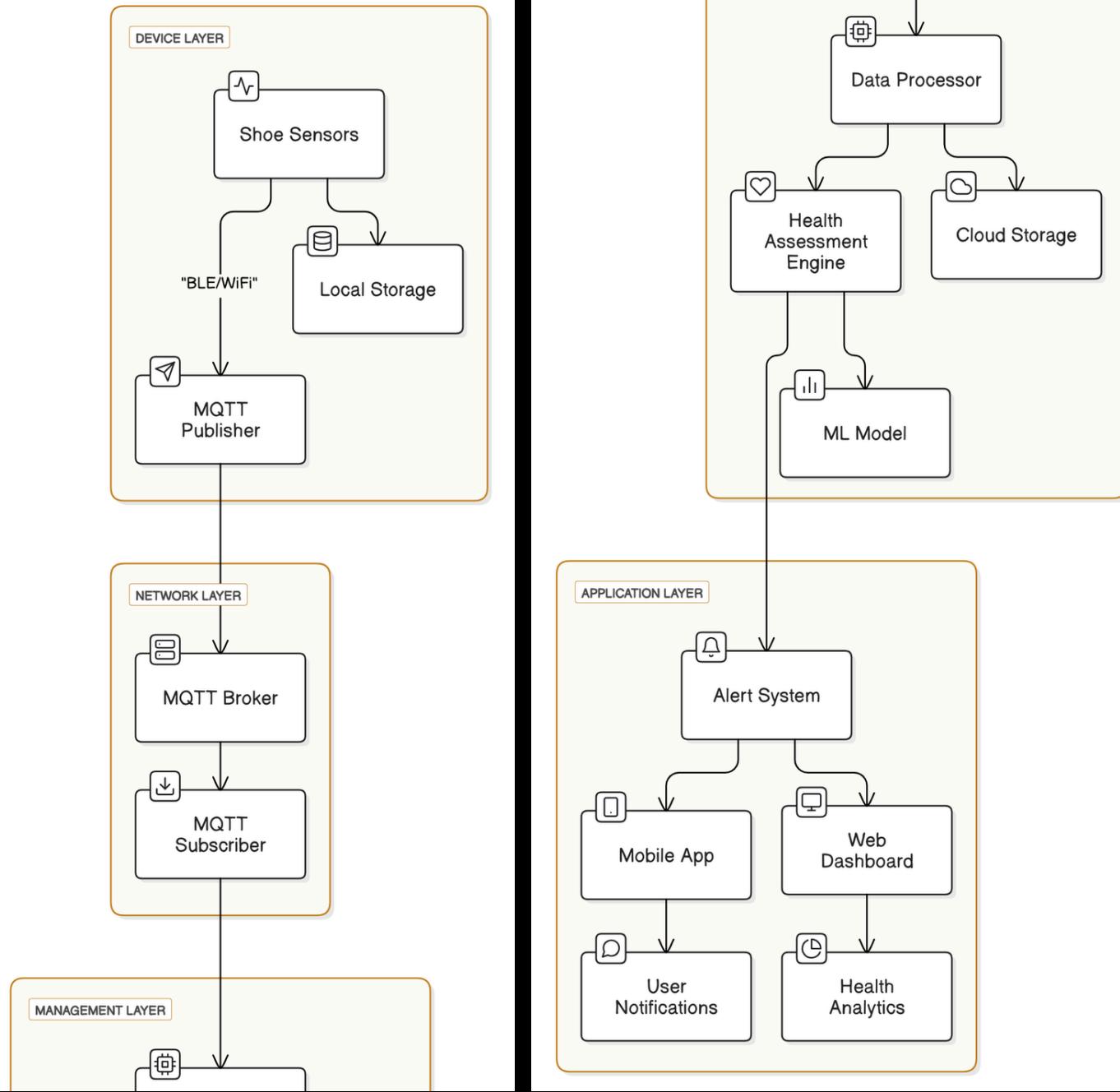


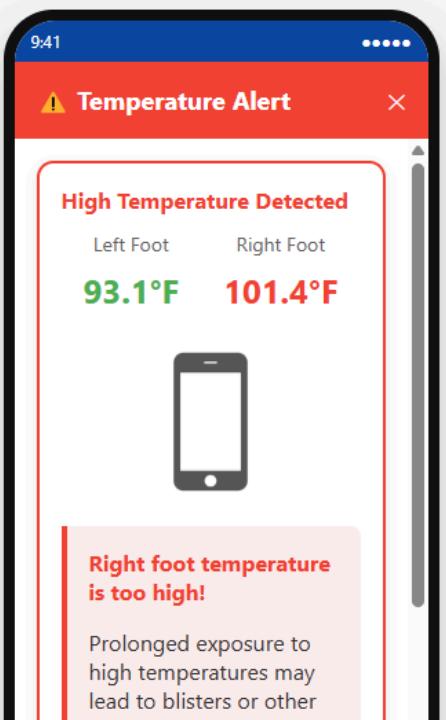
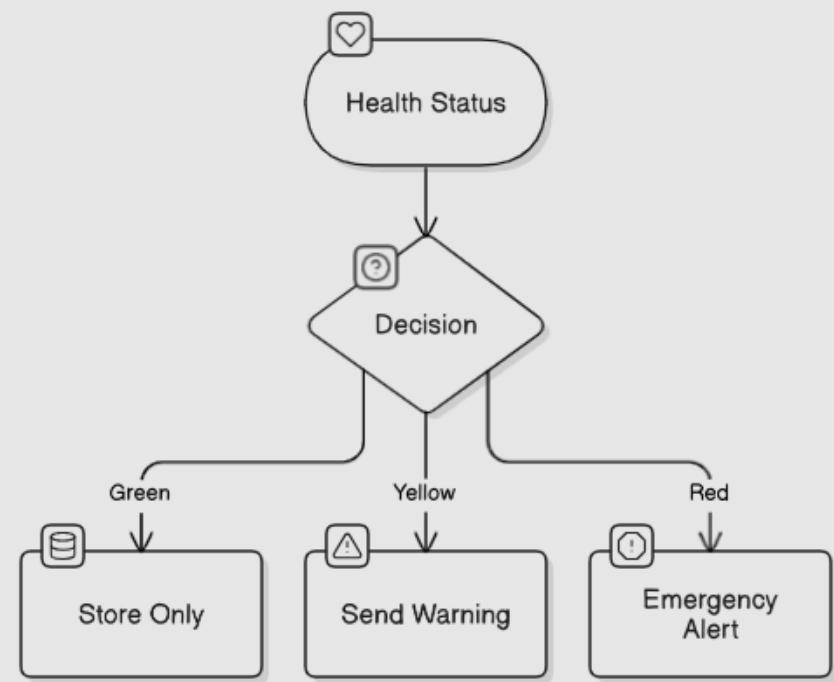
Pace

Analyzes your walking or running speed, which can help identify gait issues or areas for improvement.

By monitoring these key metrics, Lacey Pacey helps you proactively manage your foot health and prevent potential problems before they become serious.

Lacey Pacey System





Foot Health Insights

Healthy (Green)

Sensor data indicates your feet are functioning optimally, with no signs of strain or potential issues. This green indicator signals that your foot health is in a good state.

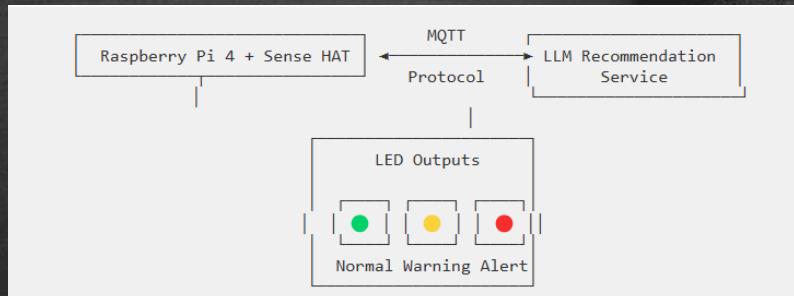
Caution (Yellow)

The sensor data suggests potential minor issues or signs of strain in your feet. This yellow indicator alerts you to be more mindful of your foot health and consider taking preventative measures.

Potential Injury (Red)

Based on the sensor data, this red indicator signifies that your feet may be experiencing significant strain or are at risk of injury. This is a clear warning to take immediate action and consult a healthcare professional.

Prototype



Hardware Components

Raspberry Pi 4

The main computing unit that processes temperature data, controls the LEDs, and communicates with the MQTT server.

Specifications:

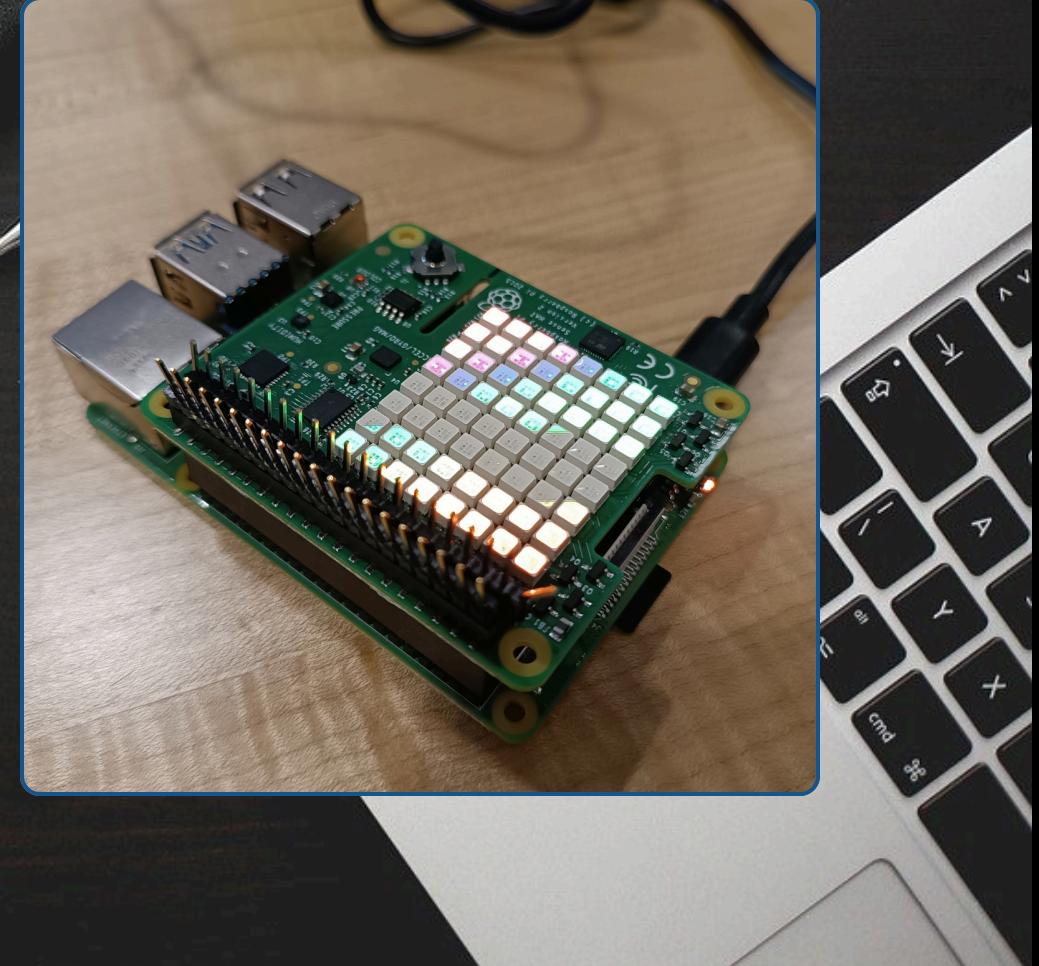
- Quad-core Cortex-A72 CPU
- At least 2GB RAM
- Raspberry Pi OS (formerly Raspbian)

Sense HAT

An add-on board for the Raspberry Pi that provides various sensors and an 8x8 LED matrix display.

Used Components:

- Temperature sensor
- LED matrix display (for alerts and messages)



```
perature_sensor_pub.py 1 ●  llm_advice_sub.py 1  
ers > thisi > OneDrive > Duke - DTI > DESIGNTK 531 > temperature_sensor_pub.py > ...  
  
import time  
  
sense = SenseHat()  
sense.clear()  
  
MQTT_BROKER = "broker.emqx.io"  
PUB_TOPIC = "laceypacey/temperature"  
  
TEMP_NORMAL = 25  
TEMP_WARNING = 30  
  
def get_calibrated_temp():  
    cpu_temp = float(open('/sys/class/thermal/thermal_zone0/temp').read())  
    ambient_temp = sense.get_temperature()  
    return ambient_temp - (cpu_temp - ambient_temp)/1.5  
  
client = mqtt.Client()  
client.connect(MQTT_BROKER, 1883, 60)  
client.loop_start()  
  
try:  
    while True:  
        temp = get_calibrated_temp()  
        print(f"Current foot temperature: {temp:.1f} degrees C")  
        if temp > TEMP_WARNING:  
            client.publish(PUB_TOPIC, f"High foot temperature: {temp:.1f}")  
            sense.show_message("HOT!", text_colour=[255,0,0])  
        elif temp > TEMP_NORMAL:  
            sense.show_message("Warm", text_colour=[255,255,0])  
        else:  
            sense.show_message("Normal", text_colour=[0,255,0])  
        time.sleep(10)
```

```
ers > thisi > OneDrive > Duke - DTI > DESIGNTK 531 > llm_advice_sub.py > ...  
  
import paho.mqtt.client as mqtt  
from sense_hat import SenseHat  
import time  
  
sense = SenseHat()  
sense.clear()  
  
MQTT_BROKER = "broker.emqx.io"  
PUB_TOPIC = "laceypacey/temperature"  
SUB_TOPIC = "laceypacey/advice"  
  
def simple_llm_response(message):  
    """Simulate an LLM by generating a recommendation based on the message."""  
    if "High foot temperature" in message:  
        return (  
            "Advice: Your foot temperature is high. "  
            "Please remove your shoes, rest, and apply a cool compress. "  
            "If discomfort persists, consult a healthcare professional."  
        )  
    elif "Warm" in message:  
        return "Advice: Your foot temperature is slightly elevated. "\\\n            "Take a short break and stay hydrated."  
    else:  
        return "Advice: Foot temperature normal. Keep up the good work!"  
  
def on_connect(client, userdata, flags, rc):  
    print("Connected to MQTT broker")  
    client.subscribe(PUB_TOPIC)  
  
def on_message(client, userdata, msg):
```

Thoughts? **Questions?**

