Assignment 2 Specification

Networking using Connected Devices

Produced by: Frank Walsh

Computer Systems and Networks

HDip Computer Science 2025

Context

- 40% of your overall mark
  - Project ethos (1 overall project)
- For this assignment you are required to:
  - Propose a project.
  - Create a working project using networking/IoT standards & protocols.
  - Typically includes a "physical" aspect – sensors/devices(SenseHat)
  - Present and communicate your work in a clear manner.
- Project will be assessed on it technical (e.g., features) content, complexity, applicability to domain and execution.

Requirements:

"Include the different layers (sensors, processing,Network/Internet, application) in an IoT Project."

Concept	Propose a domain-specific solution that addresses a problem (e.g. health care, smart home)
Apply	Apply suitable computer networking protocols and standards.
Model/ implement	Model/implement a "prototype" solution to your proposal Use the knowledge, skills and practices from other modules **Should be scoped correctly – you're not expected to build a production standard solution in a few weeks.**
Present/ curate	Present/curate your project Create a short video that demonstrates the project.

Proposed Timeline

Nov 28th

Proposal document

- Doesn't need to be exact/long!
- Sections:
  - Introduction
  - Proposed technologies (protocols/devices/prog lang)
  - Proposed Tools (IDEs, cloud platforms)
- **Deliverable: One page pdf or markdown document with general concept.**

Jan 5th

Presentation and interview:

- Git Repository submission(all code/resources)
- Include a **max 5-10 min** video demoing your project. (YouTube video)
- 10 min Zoom Interview with me in January
- **Deliverable: Accessible Github repo. Link to video.**

Proposal Document

[redacted]
Network and IOT Application.
Git Repo:
https://github.com[redacted]_Application_Assignment

I would like to restrict the time my son spends on his PS4. My proposal is as follows:

Using the Mac Address / IP Address of the PS4 which is on my home network when powered on, the Raspberry pi will detect when the PS4 is powered on, notify me through my web app to my mobile phone, start a countdown timer to 60 minutes. The sense hat will count down (in green) the minutes using the led's and when that 60 minutes has elapsed, the sense hat lights will turn red and tell the user via Bluetooth speaker to vacate the vicinity. The camera / motion sensor will be set so that after 2 minutes, if there is someone in the room, I will receive a message stating STILL GAMING!. If there is no movement, I will get a message saying ALL CLEAR.

PROJECT NAME
SHOWER SAVE
10/11/2021

STUDENT NAME:[redacted] **STUDENT ID:**[redacted]

1. Project Background and Description

*The **shower save** system is a shower length measurement and alert system. This system has two possible purposes:*

Use 1 - The purpose of this device is to notify someone by text and alarm that a person has been in the shower longer than expected. This device is useful for the guardians of anyone elderly or infirm to notify them by text or alarm, if the user has fallen in the shower and not exited.

Use 2 – This shower alarm could be used to measure the length of time spent in a shower in a country where water restrictions are in place.

Shower Save also measures shower lengths and records these times on a dashboard.

1. *Raspberry Pi is placed on the outside of the shower door.*
2. *When the door opens the raspberry pi sends an event 1 which also contains temperature value (cold) – Person enters shower*
3. *When the door opens next time, the raspberry pi sends an event 2 with temperature value(hot) – Person exits the shower*
4. *The length of time between shower start events (door opens cold temp) and shower end events (door open hot temp) is measured over a period of time and saved to dashboard on phone app**
5. *Write code to calculate the length of time between event 1 (cold) and event 2 (hot), if it is 5 minutes longer than average shower time, send text message to guardian/homeowner and trigger alarm.*

**How will the raspberry pi know that it is the start of the shower versus the end of a shower. Only measure between an event 1 where the temperature is lower (Shower not on yet) and event 2 where the temperature is higher (shower is complete).*

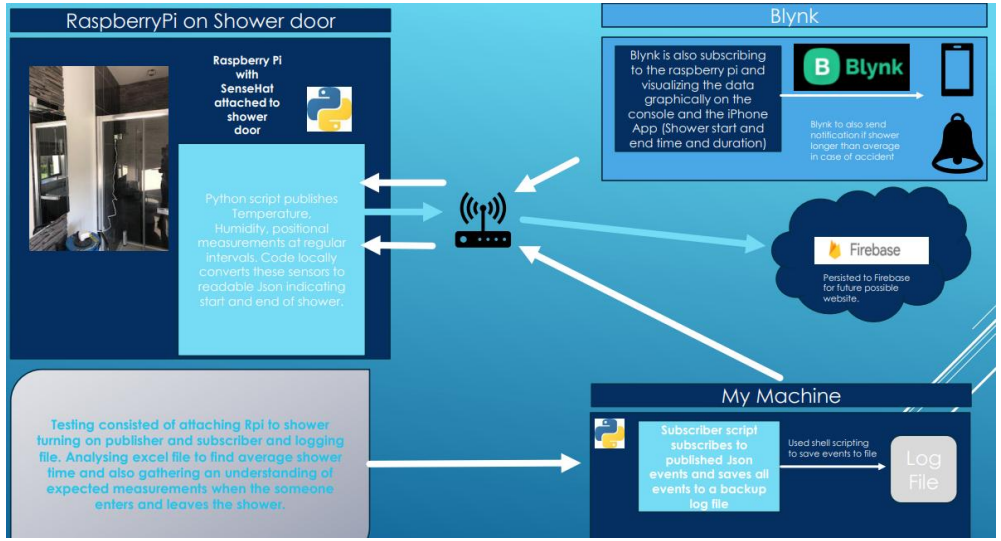
Notifications sent in the following scenario:

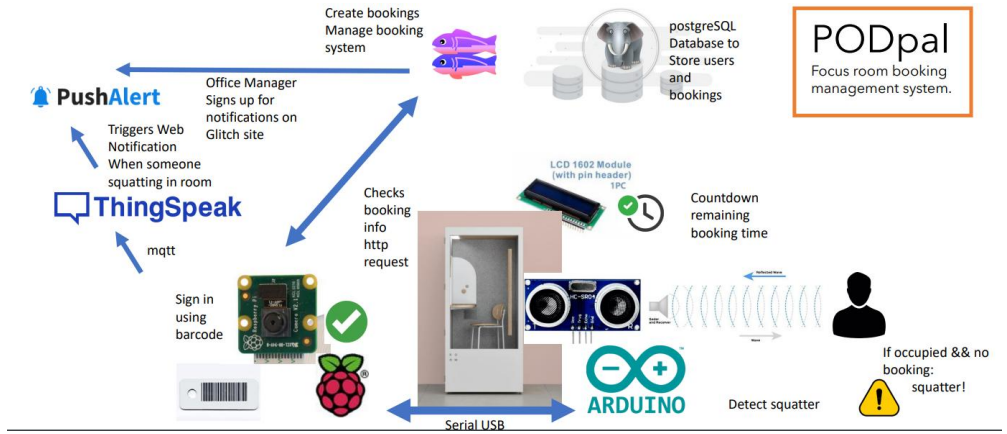
If Event 2 (high Temp) is longer than the average shower time (Time between events with low temp to events with high temps)

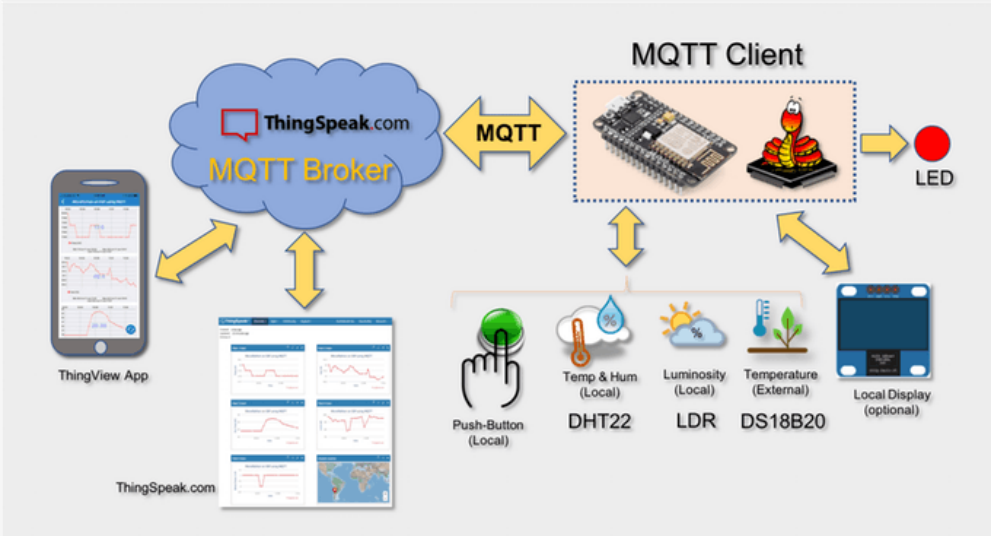
Alarm (either Phone App or Alexa)

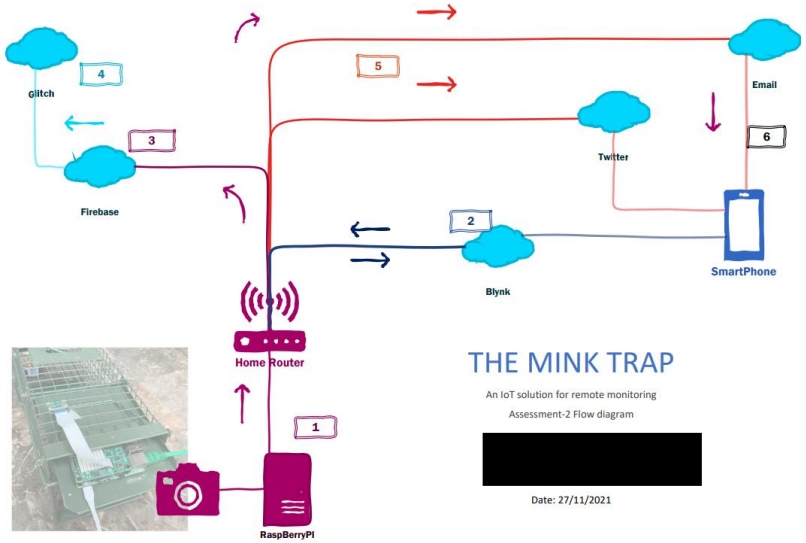
If Event 2 (high Temp) is longer than the average shower time (Time between events with low temp to events with high temps)

Project Graphic









Final Project Submission

[redacted]rfoods Public

Watch 1

Code Issues Pull requests Actions Projects Security Insights

master 1 branch 0 tags Go to file Add file Code

[redacted]ic.html 989c95b on Dec 20, 2021 43 commits

ChamberFoods	Commit changes 18/12/2021	11 months ago
assets	Add files via upload	11 months ago
website	Update chamberpic.html	11 months ago
README.md	Update README.md	11 months ago

README.md

Project Overview

Chamber Foods is an IoT project combining smart appliances, artisanal foods, and DIY.

About

No description, website, or topics provided.

Readme
0 stars
1 watching
0 forks

Releases

No releases published

Packages

No packages published

Languages

Python 69.4% C 22.0%
HTML 7.4% Other 1.2%

Grade Spectrum

	IoT Solution	Networking/IoT Technologies Used	Combined knowledge	Communication
Release 1 Base (30-49)	Basic solution that may form basis of overall application. Sensor/device focused.	Physical/Data link layer solution. Basic one way connection between device/processes.	2 programme strands present in output. Basic knowledge of each exhibited. (e.g. programming, database, computer systems)	Zip file and/or basic Repo: Minimal (1) communication resource used (e.g. simple readme.md) and video.
Release 2 Good (50-64)	Solution with clear IoT and domain application. Includes some data processing and/or gateway function.	Wireless/Wired protocols including network and transport layer. >1 protocol. Interconnected device(s) and or processes.	Apply and combine concepts from more than two modules/strands..	Good Github Rep: Repository includes clear structure, documentation.
Release 3 Excellent (65-80)	IoT Application of good prototypical standard. Used to evaluate overall suitability for a full project.	Lightweight messaging. Network/API programming. Architecture/ IOT Framework that mediates between devices and components.	>2 strands as above including more advanced knowledge and concepts.	Excellent Github Repo: Additional communication resources (e.g. instruction video, learning resources, installation guide)
Release 4 Outstanding (80-100)	Novel solution of clear applicability to specific domain. **Could result in employment/internship offer.**	All previous to excellent level. Excellent Use of Cloud/IoT specific platforms	All above, including self-acquired knowledge over and above module content.	All the above to excellent level, accessible project platform (e.g. web site, Social Media)

What Releases might look like:

Release 1(Core)

- **At least one sensor or simulated data source (e.g. temperature, button, motion, JSON generator).**
- **Regular data collection.**
- **Some local processing of raw values (e.g. thresholds, states, simple rules).**
- **One running program that shows behaviour clearly (console output acceptable).**
- **Basic logging/display (print values, write to file, simple terminal UI).**
- **Clear project description: problem, aim, and what you're building.**

“My RPi reads sensor values every few seconds, checks if it's above a threshold, logs it, and shows a warning on screen.”

What Releases might look like:

Release 2(Good)

- **Two distinct components/processes** (e.g. sensor/edge node + service/dashboard).
- **At least one network connection** (MQTT, REST API, WebSocket, TCP/UDP, HTTP).
- **Structured data messages** (JSON recommended).
- **Meaningful data handling** beyond raw logging (averages, states, or simple analytics).
- **Basic dashboard or visualisation** (web page, terminal UI, or simple graph).
- **Initial architecture diagram** (boxes + arrows is fine).

“Edge device sends JSON readings to a second service via MQTT/HTTP. The service logs the messages and shows a simple dashboard page.”

What Releases might look like:

Release 3(Excellent)

- **Multi-node or multi-service architecture** (device → service → UI).
- **Clear message schema / API contract** documented in README.
- **Historical data handling** (SQLite/CSV with queries, charting, trend indicators).
- **Live updates** (MQTT subscription, WebSocket updates, or auto-refresh UI).
- **Derived metrics or smart behaviour** (alerts, recommendations, rules engine).
- **Refined UI** showing both current and historical data.
- **Clear, well-explained architecture diagram** (separation of concerns).

“My device publishes readings to an MQTT broker. A backend service stores them and exposes API endpoints. A web dashboard graphs trends and shows alerts when readings exceed limits. . If <some event> occurs, application does <x>””

What Releases might look like:

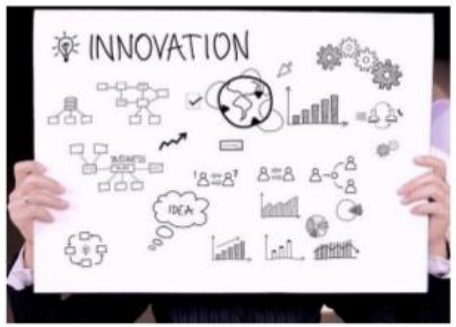
Release 3(Outstanding “stretch” features)

- **Cloud or remote deployment** (Azure IoT, a VPS running your broker/UI, or remote database).
- **Multiple communication methods** (e.g. MQTT + REST, or device → broker → cloud).
- **Configurable system** (thresholds, settings editable via UI or config file).
- **Robustness features** (reconnect logic, message validation, retry/backoff).
- **Advanced UI** (charts, filters, zones, device status, colour-coded alerts).
- **Self-learned technology** beyond module basics (Docker, containerisation, cloud services, DevOps, message schemas like Protobuf, etc.).
- **Insightful reflection** in video: trade-offs, architecture choices, limitations.

“My sensor node publishes to an MQTT broker on a cloud VM. A backend API ingests data into a cloud DB. A deployed dashboard visualises long-term trends, alerts, and device status. Thresholds can be changed from the web UI.”

Sample Project Ideas

- **WeatherTop3.0?**
  - Extend/modify WeatherTop from last semester to connect/use your own environment device.
  - Already getting current/forecasted weather conditions. Combine with home environment data to analyse and/or control stuff(e.g. rain warning – take in the washing! Clear/Windy – hang out the washing!).
- **Smart home/Monitoring device:**
  - Access and control a home device
  - Monitor/analyse home environment (medical/care apps)
- **Who's there?:**
  - Detect and monitor devices on the home network(enhance lab work with cloud platform, extend to general smart home solution, use camera…)



Other Considerations

- "Permissionless Innovation"!
  - You don’t have to limit yourself to SenseHAT/RPi or module technologies. You can integrate other devices/ systems
  - We used Python but you can use any other programming languages/ frameworks if you want. (Remember, the RPi can run Node/Java/JavaScript)
- Have a look at other IoT project examples for inspiration.
- It's OK to "simulate" sensors
  - Use Packet Tracer
    - We realise you don’t have a lab at your disposal. You can simulate data (e.g. GPS location.)