Project Sequoia

An IoT In-Home Health Care Service

[1]

Advisor:
Daji Qiao

Client
Andrew Guillemette

Team:
sdmay19-36

Cody Brooks
Guan Lin
Joshua Lang
Joshua Hatton
Justin Somers
Michael Ludewig

Problem

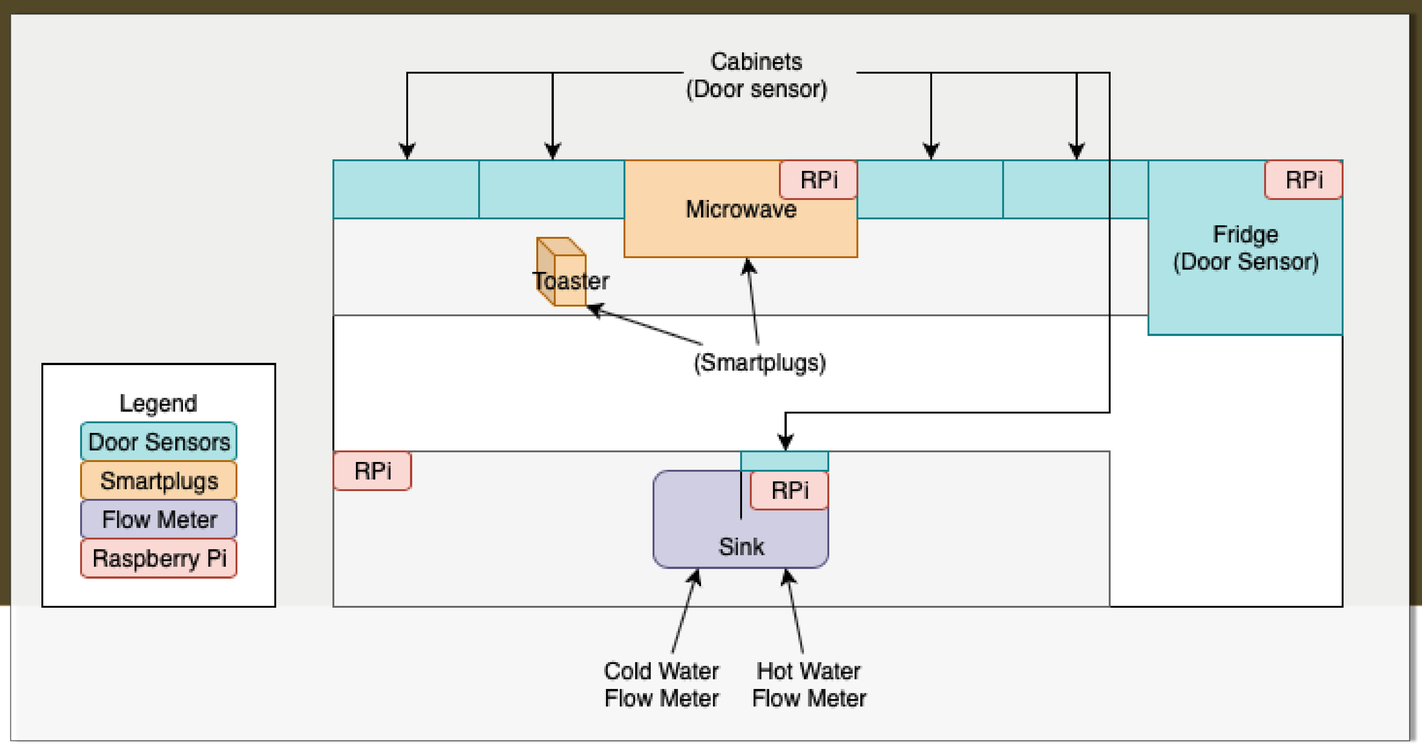
To examine a tree's health, one must examine the roots. In the same way, digging deeper into a person's habits can give insight on their overall health. That is why our project is codenamed "**Sequoia**" to signify digging deeper to get a better reflection on the health of people.

When a patient is sick, they tend to go to the doctor. The doctor will ask the patient questions to determine if something is wrong. For senior citizens, this can often be too late to get the best care possible. What if it was possible to provide healthcare professionals information about senior citizen habits and health trends?

Solution

To monitor the patient's habits Sequoia puts passive monitoring sensors in their homes. The data that these sensors collect is then used to form a behavioral profile. The behavioral profile is used to identify trends in the patient's daily habits and general health. Our group has extended an existing project by adding smart plugs, a flow meter for the kitchen sink, and a smart watch to the sensor system.

Concept Diagram



System Components

Smart Watch

- Monitor status of patient to ensure they are in good condition.
- Logs heart rate, daily step count progress, and time of event periodically and transmits to the AWS backend.
- Fossil Smart Watch running custom Android WearOS Application utilizing the heart rate sensors and step count sensors.

Android App

- Allows for doctors, family members, and caretakers to view each sensor's data.
- Requests data about specific sensor types from the database. Processes and displays received data for the user.
- Android App utilizing Volley HTTP Library

Sensor Hub

- Accepts events on local network and processes and transmits data to AWS backend.
- Raspberry Pi running NodeJS script & local HTTP server

Smart Plugs

- Utilizing the smart hub, outlet sends events when device connected begins to draw power, and stops drawing power.
- Data collected via NodeJS script subscribing to external TP-Link API, then transmit to hub.

Flow Meter

- Flow meter sends out events for hot and cold water usage.
- Meter is wired to a raspberry pi, which triggers events that are sent wirelessly via a HTTP python service.

AWS Backend API

- Accepts incoming data and processes events to compute corresponding time duration of events.
- Java Spring Backend API

Functional Requirements

Smart Plugs, Flow Meter, and Smart Watch Monitoring

Utilizing a Smart Hub, daily health events, such as using microwave, getting a cup of water, and smart watch analytics are delivered to a data analysis server.

Data Analysis Server

An analytics server accepts incoming health events and distributes the information to the various systems using it (Mobile App, Webpage), while analyzing the data and providing helpful information to the user.

Non-Functional Requirements

Scalability

3rd Party products used are readily available for purchase in large quantities to be scaled to large amounts of customers.

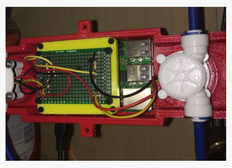
Wireless

Implemented systems operate wirelessly to transfer data and communicate with the smart hub and user.

Independent

The product requires no user interaction or maintenance to continue monitoring and sending data. Accepts updates remotely.

Flowmeter Internals



TP-Link Hs110 Smart Plug

Installed Flowmeter

Fossil Gen 4 Smartwatch

System Architecture Description:

Sink Usage

Sends data on whether hot or cold water is used and how much water used.

Device Stats

Information on when a device uses power and what time it is used, collected from the smart plugs.

Drawer Open/Closed

Sends info on when a kitchen cabinet drawer is open and for how long.

Sensors/Smart Plug Data

The sensor hub sends all data from sensors and plugs to server through Wi-Fi.

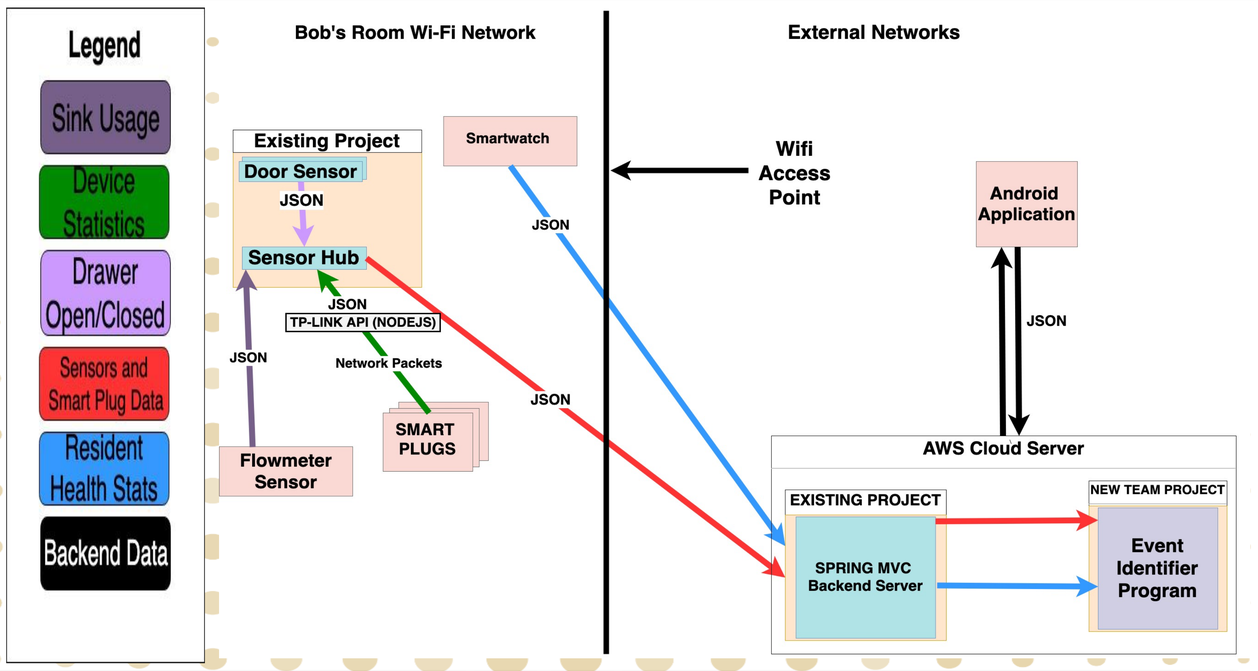
Resident Health Stats

Info on senior citizen's heart rate and step count collected from an app on the Fossil smartwatch. Data sent through Wi-Fi.

Backend Data

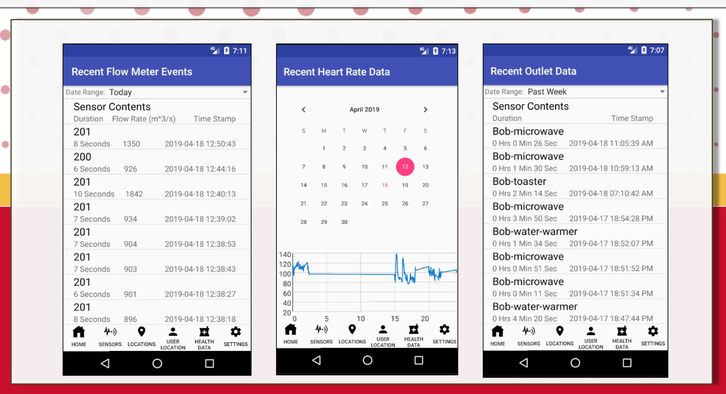
Data that is exposed from the AWS server to the Android app.

System Architecture & Design



Conclusion

Our implementation of smart plugs, flow meters, and the smart watch are used to collect data. The android app is used to display that data will create events for a behavioral model to track patients' health. In the future, more sensors can be added to this system to track and display the new sensor data on the app we developed. Another future goal would be to use the data & machine learning to predict different events, such as eating or making coffee.



Testing

Smart Plug

Set up a development environment and test environment in the pilot program so developers could test changes locally first, then send to development environment for the whole team to test, before deployment to the live system. During deployment, monitored data closely to make sure it matched with real life events.

Flow Meter

Hooked the meters up to a self-built water tank to simulate the flow of water to ensure the flow data tracked matched the amount of water that came out of the tank. After passed all tests in this stage, flow meter was deployed to live system.

Smart Watch

Watch was worn by team members to ensure data was correctly sent and received. After successful testing, device was handed off to Bob Kern to test in the live system.

Acknowledgements

We would like to acknowledge Bob Kern for agreeing to be part of the pilot program for our system and allowing us into his home to install and test different aspects. Also Green Hills Retirement Community for their support in allowing us to test in their facility.

[1]: http://clipart-library.com/clipart/1712342.htm