Home Health Monitoring System

2020-12-04

Ziyad Allehaibi, zlehaibi
Terry Edwards, tedwa98
Ben Whalin, vaeca21
Elijah Rose, elirose
Mentor: Gregory Myers (gmyers), gmyers
Instructor: Abdollah Mirbozorgi, samir

Abstract

A Home Health Monitoring System to encourage extensibility of monitoring devices in care-at-home applications.

A Home Health Monitoring System to encourage extensibility of monitoring devices in care-at-home applications.

Table of Contents

Introduction

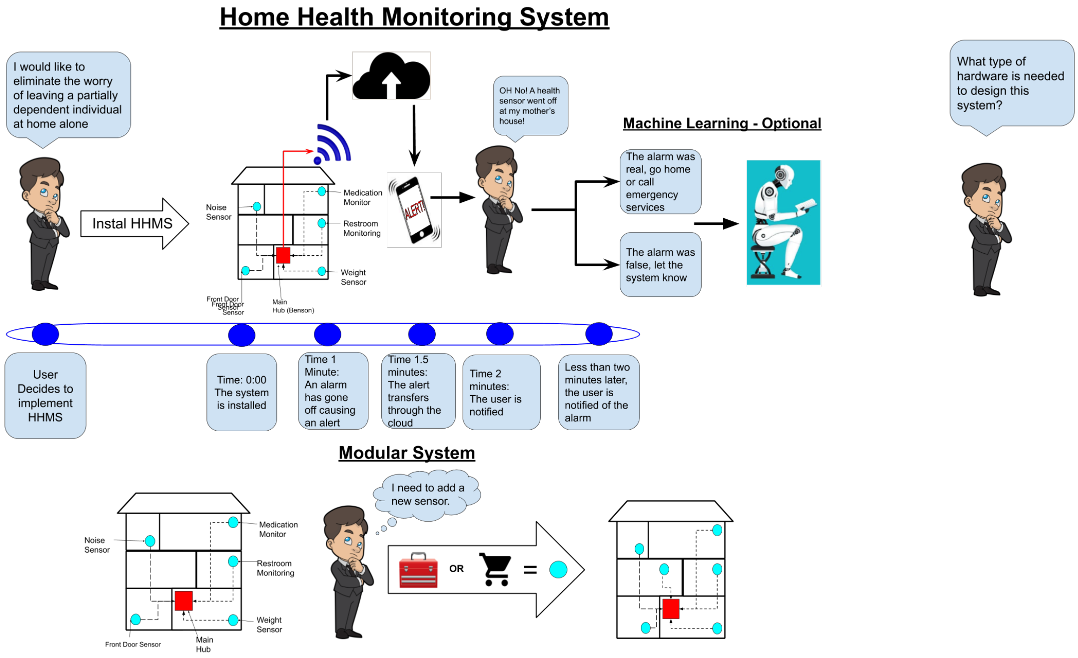

Figure 1: Top Level Diagram

Background

Existing Solutions

Project Description

In order to address the aforementioned problems, the team proposes an open-source home health monitoring system with an API to encourage extensibility to third-party devices. The system, initialized as **HHMS**, seeks to create and focus on a framework moreover than a single implementation, allowing relatively simple additions to the system without necessitating a recreation of the multi-step communication setups and data processing.

The project thusly focuses on establishing this framework and its API, allowing near any device to integrate with the system given it adheres to the protocols set by the system.

Sample

While the framework is the main focus, it is dificult to show that a framework exists without a working solution, and indeed multiple. If the system only demonstrated a single communicating device, it would hardly justify or prove the extensibility. Hence several sensor sub-systems will be set-up to work with the framework as an example of its capability and proof-of-concept.

Capabilities

- **Data Acquisition and Communication**: the user shall be able to read the data from their sensor subsystems, as well as communicate back to them in order to set their properties.
- **Data Processing**: data shall be processed according to set rules in order to indicate discrepencies with expected behavior.

Machine Learning

In order to further aid the efforts of the caretaker, behavior anomaly recognition should be implemented in order to detect irregular patterns or sharp steps in established patterns. The implementation will allow short-term recognition of emergencies (e.g. falls) as well as unhealthy longer-term patterns (such as a sudden drop in weight over a few days).

Design Process

After establishing the main project goals, design began with the overarching view of the project and brainstorming solutions via conceptual block diagrams, as d

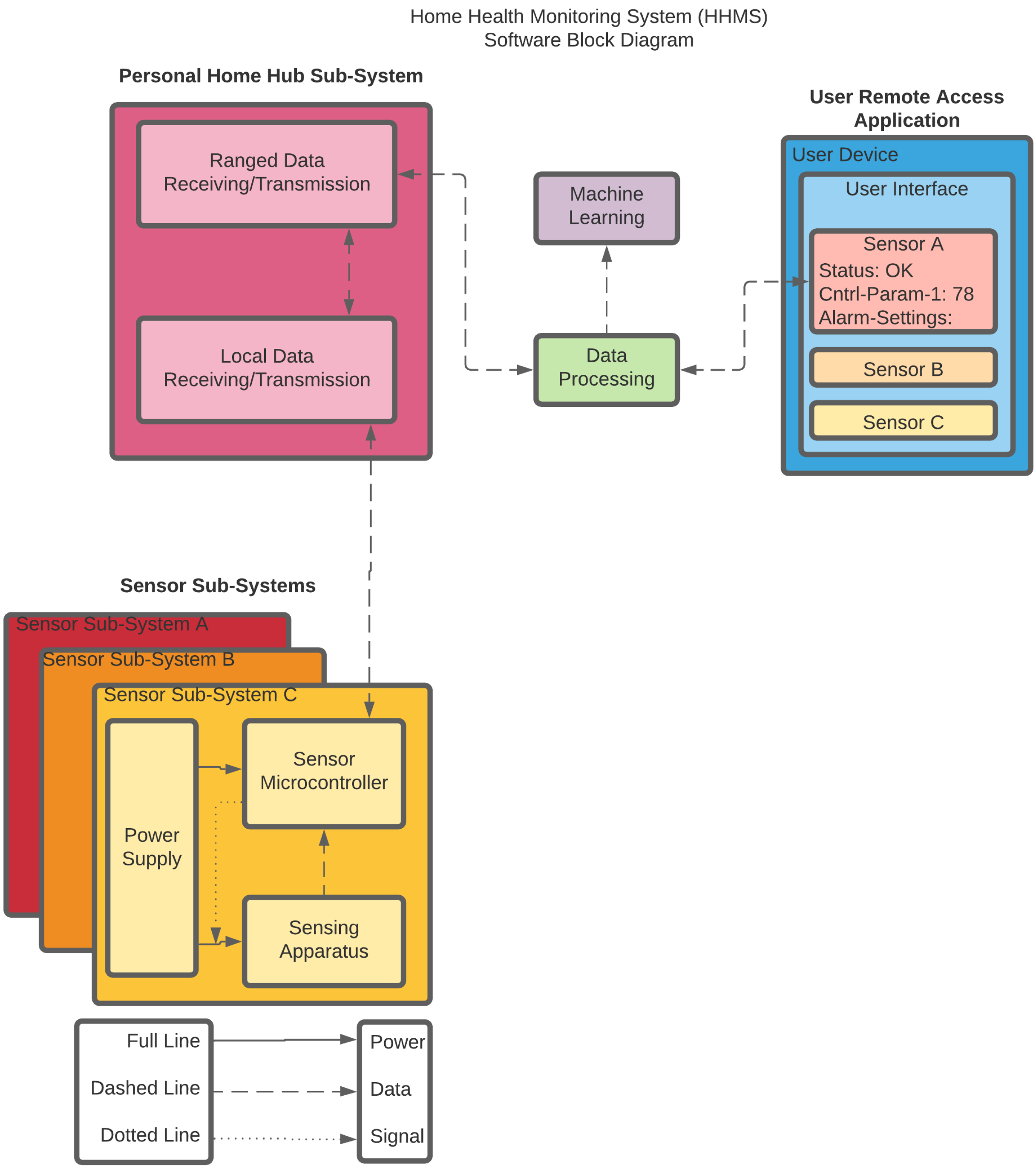

Figure 2: An early conceptual diagram for the outline of the software systems.

To this end, there is already a necessary separation in the software – data-steams of the connected devices and metadata used by the system to interpret the data-streams – as well as hardware – sensor component and system wrapping microcontrollers. Additionally, due to the emphasization on a DIY and simple extensible set-up, user communication through the system is inherent to the design.

After the broad design was established, the individual components and implementation details were established in decision tables using a calculated Figure of Merit system. These decisions were then processed into the design table that outlines the final component-level decisions. Due to the inexperience of team Engies, this system is crucial in order to determine the proper materials for the project and develop the parts list without merely relying on the limited scope of products introduced in courses.

Methods

Decision Tables

The decision tables compile the research into the relevant specifications

								Input Types
Micro Controllers	Bluetooth	Wifi	USB 1.1	USB 2.0	USB 3.0	USB C	RS-232	HDMI Type A - Standard
Weights: 0-10; Values: Boolean	0	1	0	5	5	2	1	4
	Value:	Value:	Value:	Value:	Value:	Value:	Value:	Value:
Raspberry PI 4B 4GB	1	1	0	1	1	1	0	0
Raspberry PI 4B 8GB	1	1	0	1	1	1	0	0
Arduino NANO 33 IOT	1	1	0	1	0	0	0	0
Banana Pi M3	1	1	0	1	0	0	0	1
Odroid XU4	0	0	0	1	1	0	0	1
NanoPi NEO4	1	1	0	1	1	1	0	1
UDOO BOLT V8	1	1	0	0	1	1	0	1
UDOO X86 II ULTRA	1	1	0	0	1	0	0	1
ASUS Tinker Board	1	1	0	1	0	0	0	1
Onion Omega2+	0	1	0	1	0	0	0	0
Orange Pi 4B	1	1	0	1	1	0	0	1
NanoPC-T3 Plus	1	1	0	1	0	0	0	1
Le Potato - AML-S905X-CC	0	0	0	1	0	0	0	1
Orange Pi Zero Plus2	1	1	0	0	0	0	0	1
Raspberry Pi Zero W	1	1	0	1	0	0	0	0
Intel® NUC Board NUC7i3DNBE	1	1	0	1	1	0	0	1

Figure 3: Top Level Diagram

Microprocessors	Bluetooth	Wifi	USB 1.1	USB 2.0	USB 3.0	USB C	HDMI Type A - Standard	HDMI Type C - Mini
Weight: 0-10	10	7	1	5	5	2	1	4
Values: Boolean								
Raspberry PI 4B 4GB	1	1	0	1	1	1	0	0
Raspberry PI 4B 8GB	1	1	0	1	1	1	0	0
Arduino NANO 33 IOT	1	1	0	1	0	0	0	0
Banana Pi M3	1	1	0	1	0	0	0	1
Odroid XU4	0	0	0	1	1	0	0	1
NanoPi NEO4	1	1	0	1	1	1	0	1
UDOO BOLT V8	1	1	0	0	1	1	0	1
UDOO X86 II ULTRA	1	1	0	0	1	0	0	1
ASUS Tinker Board	1	1	0	1	0	0	0	1
Onion Omega2+	0	1	0	1	0	0	0	0
Orange Pi 4B	1	1	0	1	1	0	0	1
NanoPC-T3 Plus	1	1	0	1	0	0	0	1
Le Potato - AML-S905X-CC	0	0	0	1	0	0	0	1
Orange Pi Zero Plus2	1	1	0	0	0	0	0	1
Raspberry Pi Zero	1	1	0	1	0	0	0	1
Intel® NUC Board NUC7i3DNBE	1	1	0	1	1	0	1	0

Figure 4: Top Level Diagram

Design Tables

Input Table

Output Table

Timeline

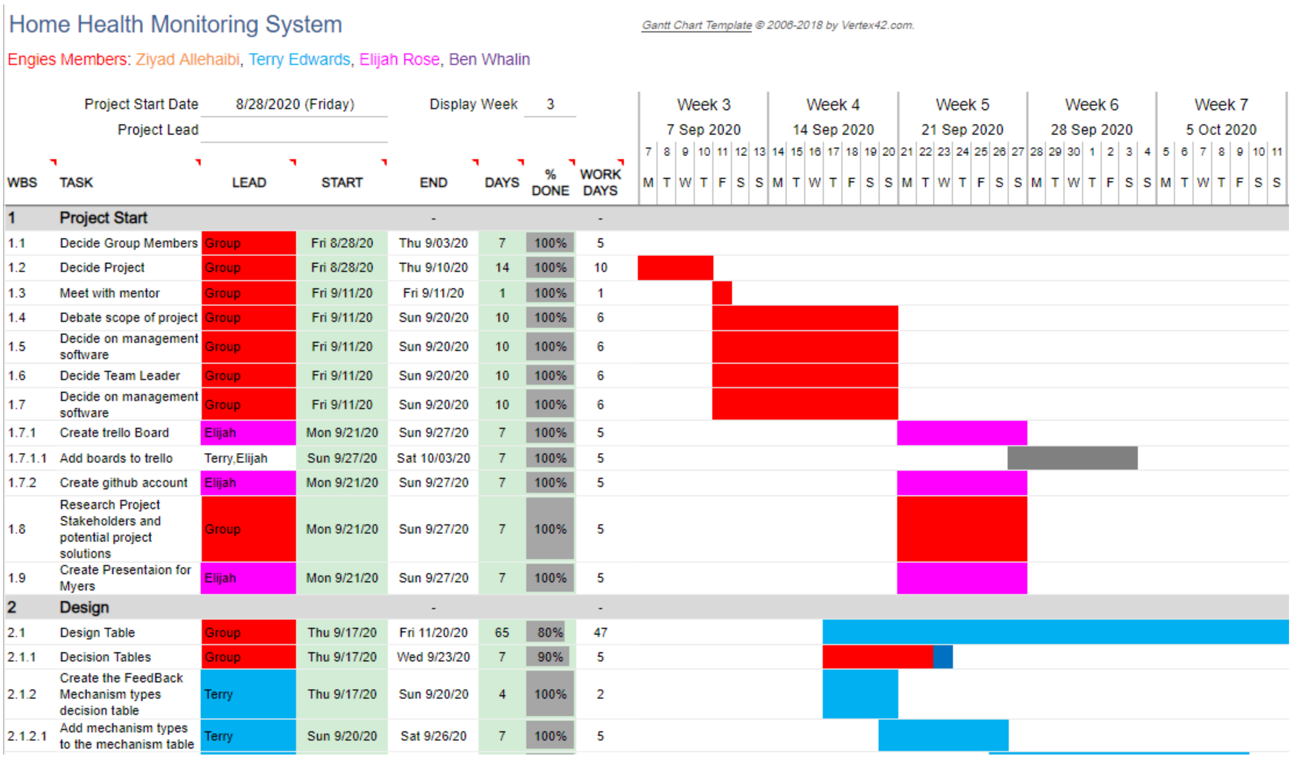

Figure 5: Gantt Chart

System Design

Hardware Design

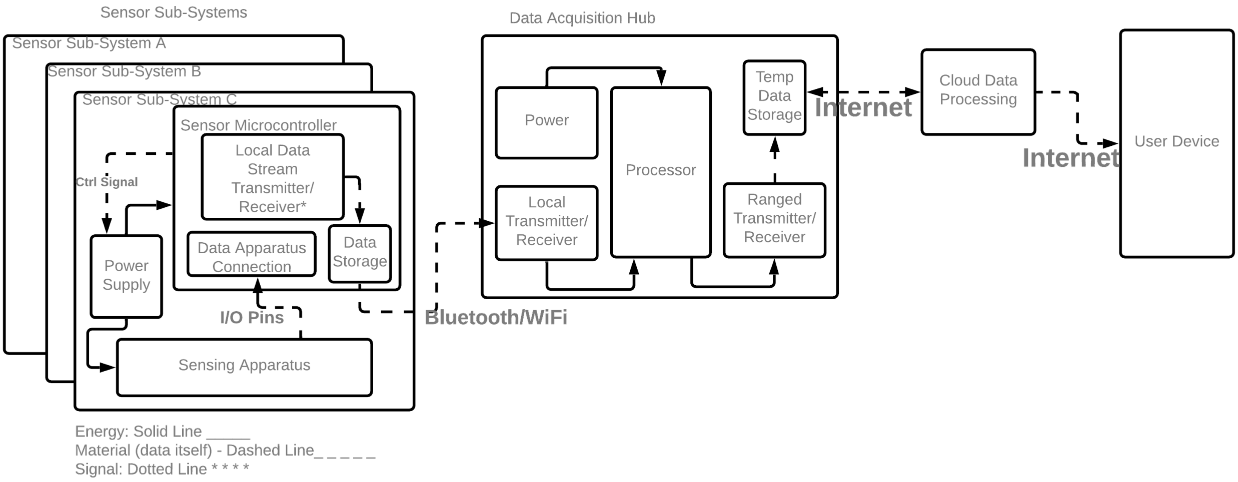

Figure 6: Hardware Block Diagram

Hardware Simulation

Software Design

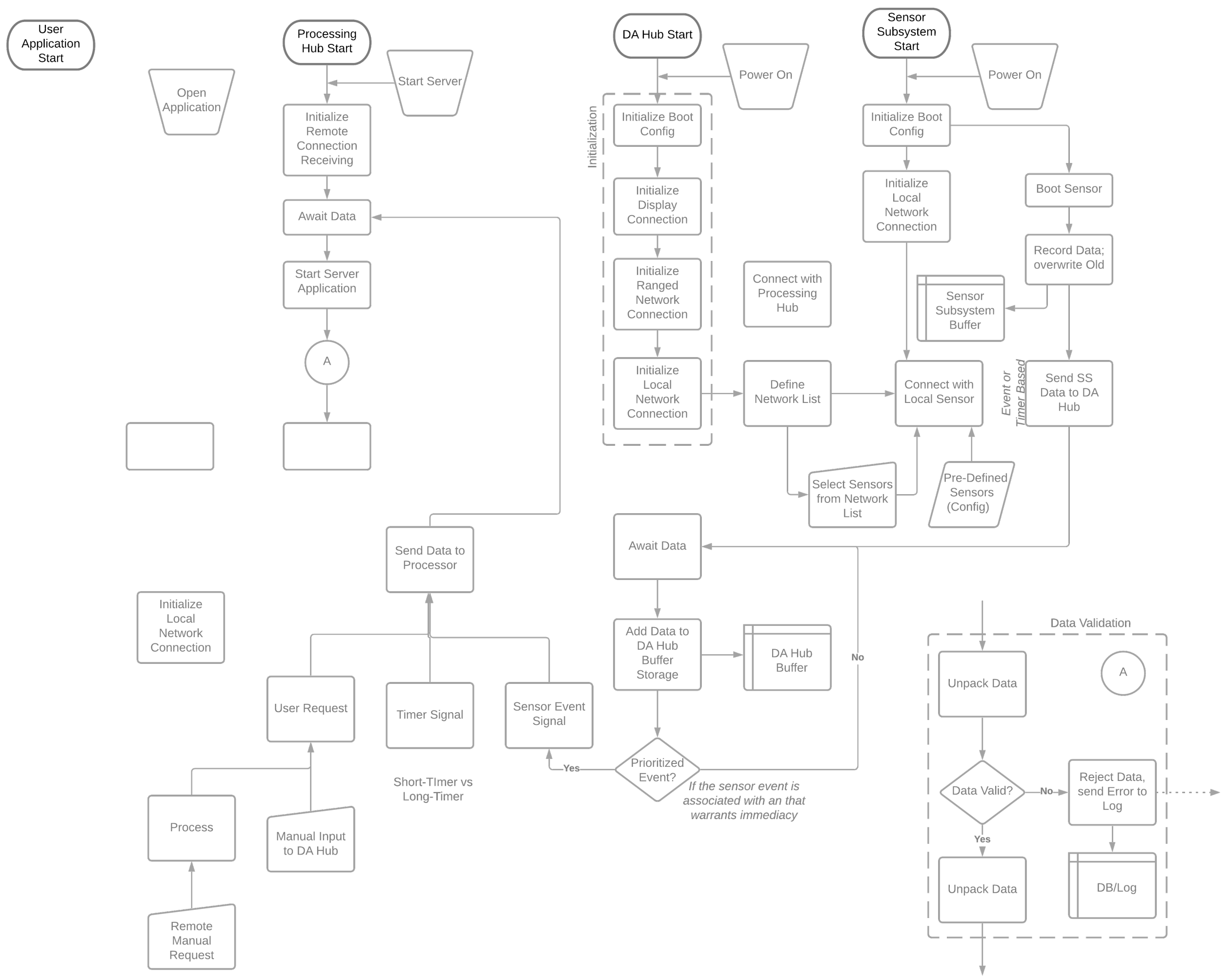

Figure 7: Software Flowchart

Software Simulation

Cost Analysis

Individual Tasks

Ziyad Allehaibi

Terry Edwards

Ben Whalin

Elijah Rose

Elijah focused primarily on the software design aspects of the project, along with the aid of Terry, as detailed in software simulation. He additionally contributed various other pieces and work to the team, such as the hardware diagram V1, separating the Data-Acquisition and Processing hub, etc.