UCF SENIOR DESIGN 1

Smart Surveillance Hub
03-25-2022

Department of Electrical Engineering and Computer Science University of Central Florida
Dr. Lei Wei

Initial Project Document and Group Identification
Divide and Conquer

Group 15

Matthew Weinert	Computer Engineer
Korey Lombardi	Computer Engineer
Kenneth Ancrum	Electrical Engineer
Joshua Sherrill	Electrical Engineer

Table of Contents

1.0 Executive Summary

Privacy and security are a standard in today's society, and the ability to keep yourself and your family safe should be accessible and affordable. According to the FBI's Uniform Crime Reporting (UCR) Program, the number of burglaries has decreased 48.5 percent from 2010 to 2019. We believe the decrease in burglaries is directly related to the increase in IoT devices. With such an increase in IoT devices, there's been a vast growth in the number of cost-effective ways to secure your home. A visible camera displayed in the open will most likely scare off a burglar who realizes they are being recorded.

Security companies tend to fully integrate homes with alarms and sensors, building a very complex system. Also, these types of systems usually won't alert you until an intruder has broken into your home and set off the alarm. By the time an intruder has entered the home, it might be too late to respond accordingly. If you were alerted before the burglar entered your home, you might have time to call the police, hide somewhere, possibly scare the intruder off, or even get ready to defend yourself accordingly.

It used to be nearly impossible to have your home secured without spending a massive amount of money on unwanted features. Nowadays, having a security company install a fully-integrated system comes with an extremely high cost, and it usually includes a contract that consists of installation fees and continuous subscriptions. For example, ADT has a similar product with more features, but it also has an extensive contract and installation fee. The system they provide doesn't allow you to use your own personal IP camera, so you have to spend extra money on their marked-up cameras. If you aren't interested in extra features that don't fit your needs or you don't want to be locked into a contract, you really just need an IP camera to solve your problem.

2.0 Project Narrative

The problem with products on today's market is they include too many features, and the prices are way too high. Our product will target the customer who is looking for a human detection product, who is looking to have a single camera or multiple cameras installed at a low cost. Since most security companies generally require you to use their cameras that are usually marked up in price and they typically charge an installation fee, we believe to have a strong market advantage. Our main goal is to build a product for the average person who understands technology and wants to buy an affordable product. Our product will be inexpensive because it won't require a technician to provide setup and installation. Our project will be modular and straightforward, making it easy to add and remove cameras from its current configuration.

2.0.1 House of Quality

A House of Quality graphic may be used to represent how our marketing and technical criteria intersect. It uses a symbol to represent where each marketing standard connects to each technical specification, indicating how strong the connection is. This may be found in the legend, as seen in the illustration below.

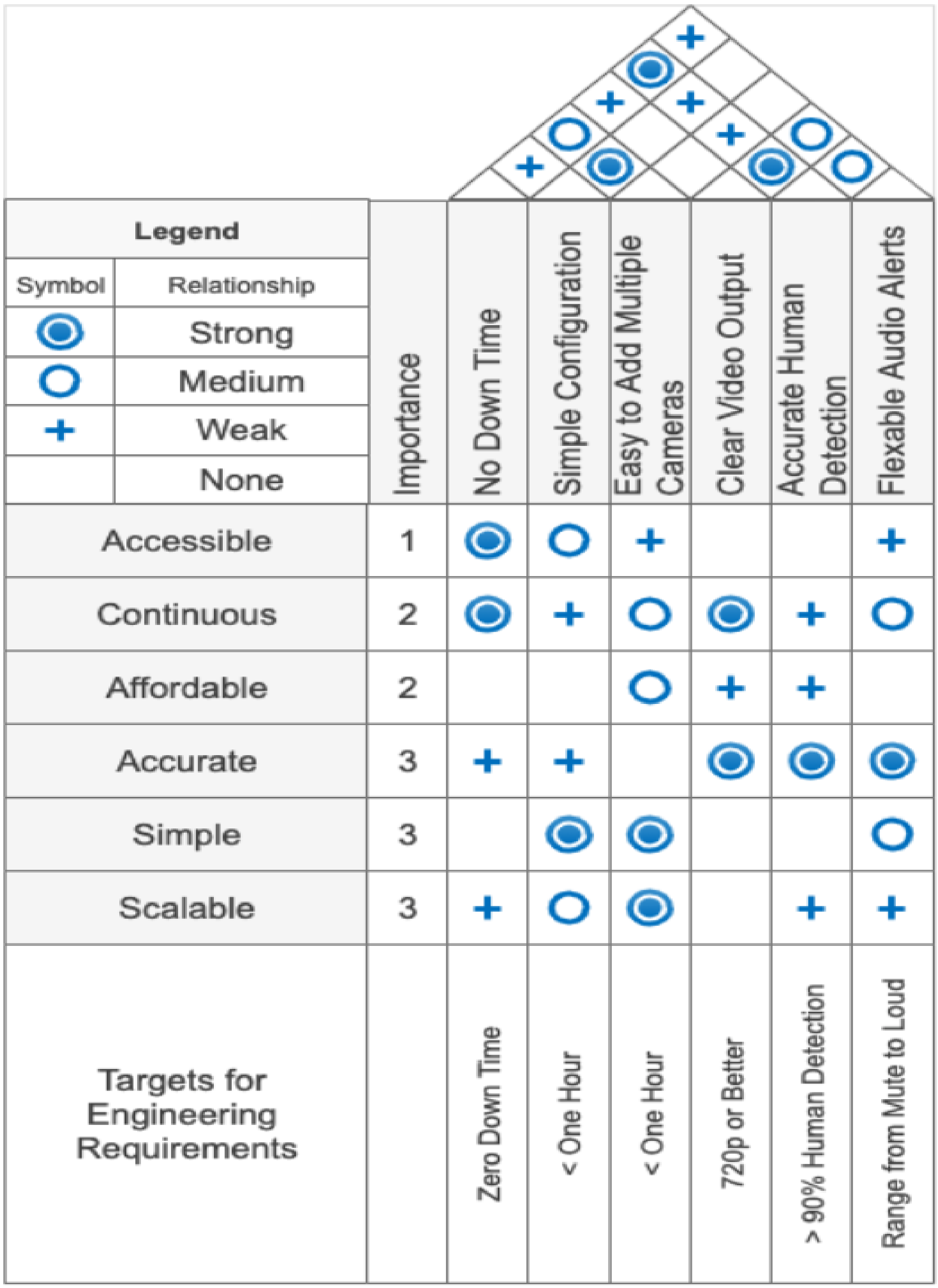


Figure 1:House of Quality

2.1 Project Objectives, Goals, Specifications, Requirements

Security Smart Hub's primary focus is to provide a high-quality, affordable, and flexible product.It will be connected through wifi to a hub that streams real-time video that you'll have access to through a clear unpixelated LCD display. Whenever the IP camera detects any human presence on your property, you will be alerted through the LCD display that's attached to the hub. The hub and display will be located inside your home, and the IP camera will send the video to the central hub to be processed. We will use facial recognition to determine if somebody is on or around your property. The different types of cameras that can be configured for use will be incredibly flexible. It will be configurable to run on the smart hub as long as it's an IP camera. This is extremely useful if you want to replace your active camera with another one due to malfunction or you simply want to upgrade to a higher quality IP camera.

Many IP cameras today only come with an app and no onsite video viewing method, meaning that if someone is not staring directly at their phone as a burglar attempts to break into their home then they will have no knowledge. With the smart hub the home owner will be notified the moment anyone comes into view of their camera via a led flashing as well as a loud noise coming from the hub. The owner can then view the hub and understand the situation. Another feature of this device is onsite storage as well, another issue with the IP cameras such as Ring, is they do not offer continuous recording storage. They offer cloud storage but it comes at a monthly subscription cost. With the Smart Hub they will be able to have their data stored onsite and be able to cycle through a month of recording to make sure no matter what happens they know their home will be protected.

By utilizing low-cost yet effective IoT devices, we will provide an affordable way of keeping a close eye on your property. Since fully integrated security systems are so expensive, we plan to save you money, whether you're a lower-class, middle-class, or upper-class individual. Surveillance Smart Hub is perfect for you if you'd like to have live footage of your property sent to you in real-time. Our product will be affordable, flexible, and simple yet very effective.

Specification	Specification Description

Number	
1	Users will have access to a Desktop and/or Mobile application to interact and view the data sent by the IP camera.
2	Smart hub’s desktop application will be able to interface with any generic IP camera within the same network.
3	An IP camera will send a continuous video feed (24/7) to the smart hub.
4	A micro-controller that is able to run the desktop application that will act as a “hub”. It will utilize A.I. to detect different objects and that information will be relayed to the notification PCB.
5	The smart hub must be able to connect and display to a monitor using HDMI to display the desktop application.
6	Parent desktop application will be able to display the live streamed video feed on a monitor that will connect via HDMI. This application will also save recorded video feed onto a storage device.
7	A PCB that will receive data from the hub and notify the user with audio on the object the IP camera detects. This PCB will also change the mood light and Volume depending on the level of security concern.
8	LCD that will be configured to the 3d printed case that will be able to run the desktop application as an alternative to using a monitor.
9	Lighting that will utilize the results of the A.I. detection to act as a security threat ‘mood light’. Depending on the level of security threat, the light will switch colors. (Red = threat, Yellow = warning, Blue = no threat)

Table 1: Specifications

2.2 Research and Investigations

The research section of this document will discuss the reasoning behind the design choices we choose for implementing our project. We'll look at other similar products to analyze their pros and cons and compare their features with the features we plan on implementing. With the research attained, we'll be able to determine the necessary functionality required to make our project a more marketable product. An in-depth analysis of different product features lets us gauge what their projects are missing that we can implement to make ours a more competitive product on the market. We will also analyze the features that have already been adopted so we can use similar concepts but with improved functionality. This section will also include extensive research on the possible hardware components, selection of parts, and relevant technologies that can be used for building our project. The primary goal of the research is to understand the technologies currently being used for building products similar to ours.

2.2.1 Hardware Research

This section is research on some of the requirements to meet the specification requirements, some different parts that can be used to achieve the requirements and go into a bit more detail on what the options are. There are various hardware parts that are going to be required in order to achieve the goals and efficiency of this project. The main components involve the LCD display, the DSI interface, LEDs, DAC, and a Power supply. Each subsection will go into detail regarding what they are and they are needed for this project.

Each component needs to be carefully selected in order to work well with the software we selected. A major concern involving the hardware is going to relate to the providing power to each part of the Hub from the computer all the way to the speakers. The basic design will involve a LCD display that is able to display and control the camera, if a person is detected using the AI face recognition software, a signal will be sent from the raspberry pi to the DAC. The DAC will then play an audio through the speakers and trigger the LEDs. Overall the hardware design seems pretty simple from an overview but there are a lot of parts that are required in order for the system to work smoothly and coherently throughout.

Regarding the hardware design there are a few options besides the ones mentioned that could have been added in order to make the project easier to complete. The more complexity added to the design of the hardware adds cost to the project. When the goal of the project is to keep the cost as low as possible.

2.2.1.1 LCD Display

Figure 2: Capacitive Touch Raspberry Pi LCD

In the Smart Surveillance Hub there will need to be a large liquid-crystal display (LCD) that is able to be easily controlled by the user. A LCD consists of 6 main components

1. A polarizing filter that polarizes light as it enters.
2. Glass substrate that consists of ITO electrodes.
3. Twisted nematic liquid crystal.
4. A glass substrate with ITO electrode film.
5. Polarizing filter film.
6. A reflective surface

We have the option to choose between two forms of LCDs: Display Serial Interface and High-Definition Multimedia Interface. No matter which interface we choose between or display it will need to have capacitive touch capabilities in order to meet our design criteria for ease of use.

For the selection of a LCD there are many options available but the goal is to find the LCD display that offers a large surface area for the user to see from a good distance away as well as being cost efficient. LCD can get expensive especially when trying to find one in order to the criteria of this project. The display has to meet the minimum requirements of the cameras we are testing in the project, the display needs to output a minimum resolution of 1080p. In order to take an invention like this to market, a LCD would probably need a slightly better resolution, but for the scope of this project and the budget limitations we are going with the minimum of our standard requirements.

2.2.1.2 Display serial Interface

Figure 3: Raspberry Pi Display Serial Interface

Since the Raspberry pi 4 module has the ability to accept a Display serial interface we considered this a display option for lowering the cost of our overall build. The DSI model was a specification introduced by the Mobile Industry Processor Interface (MIPI) Alliance,This is a global tech alliance that specializes in developing specifications for the mobile ecosystem. Using DSI is our best option for low power consumption, since the MIPI interface uses low voltage signaling which has the benefit of low power operation. But even at such a low power it can transmit at a wide range data transmission allowing for less interference of other peripheral devices while allowing for less pins and offering better performance than other interfaces.

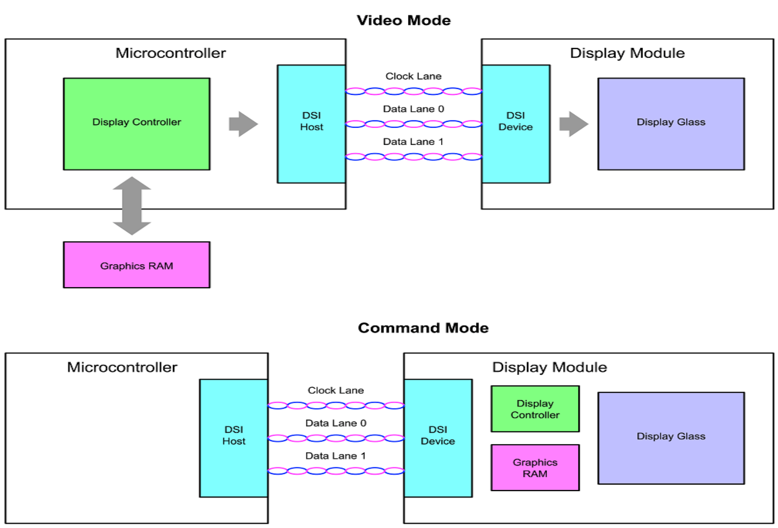


Figure 3: Hardware Communication Diagram

The interface operates with a differential signaling point-to-point serial bus on the physical layer with the latest with the latest lanes for D-PHY operating at speeds of 4.5

Gbps/lane. The Bus consists of one high speed clock lane and one or more data lanes where each lane is carried on a two wire configuration. All lanes will bridge between the DSI host and device where the lanes will transmit in parallel, meaning if four lanes are being used then all 4 bits will be transmitted simultaneously. There are two modes that can be used: the video mode and the command mode.

The video mode will be the standard operation for most systems. In Video mode RGB pixel data and horizontal and vertical sync signals provided by the display controller are encoded into the serial stream by the DSI Host (in our case this would be our raspberry pi 4 microcontroller) and decoded by the Device. Alternatively, the display controller and graphics RAM can be integrated into the display, this can take the load off the DSI host as its only responsibility would be to transmit the pixel data. These functions make the MIPI DSI interface a lot more complex than the classic parallel RGB plus clock and sync signals, but offers High performance, Low power, and Low EMI.

2.2.1.3 High-Definition Multimedia Interface

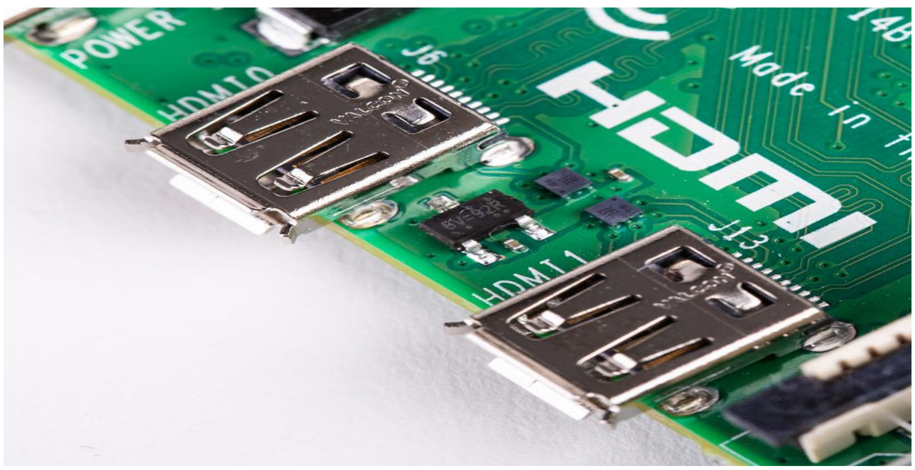


Figure 4: HDMI interface on a Raspberry Pi

The high-definition multimedia interface is currently an industry standard for video and audio transfer on LCD displays, and our microcontroller comes equipped with micro hdmi ports capable of transmitting a 4k image. The HDMI interface has replaced the old standard of analog Video standards for uncompressed/compressed audio and video transmission.The HDMI standard implements EIA/CEA-861, which defines video formats and waveforms, transport of LPCM audio both compressed and uncompressed, auxiliary data, and implementations of the Video Electronics Standards Association Extended Display Identification Data. although there are many different version of HDMI that basic output is designed with 19 pins that are sectioned off for Transition-minimized differential signaling (TDMS) data, Consumer Electronics Control (CEC), HDMI Ethernet Channel (HEC), and Audio Return Channel (ARC) , as well as power pins and I2C data lines. To be more specific, the HDMI has three physically separate communication channels, which are the DDC, TMDS and the optional CEC with HDMI 1.4 adding ARC and HEC.

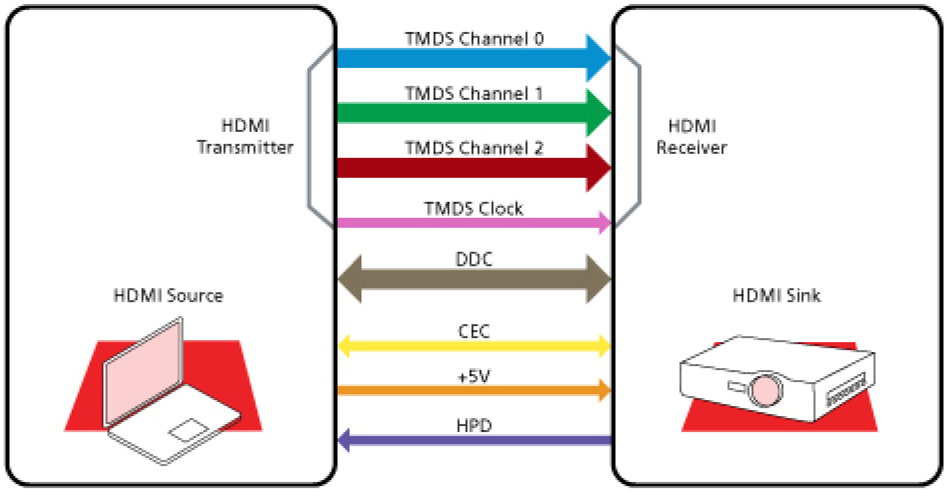


Figure 5: HDMI pin communication diagram

TMDS on HDMI combines video, audio and auxiliary data using three different packet types or channels referred to as the video data period, the data island period, and the control period. When in the video period active pixels are being transmitted, the channel is sending 10-bit characters that are encoded using 8b/10b encoding during the video period. This differs from the island period where audio and auxiliary data are transmitted into packets with each packet containing four subpackets, and each subpacket is 64 bits in size, including 8 bits of parity data used for error correction and describing the contents of the packet. This allows for each packet to carry up to 224 bits of audio data.

DDC is a communication channel based on the I2C bus specification. Specifically the Enhanced Display Data Channel would be implemented to read the E-EDID to discern between what audio/video formats the HDMI source device can take. This function is also necessary for being used for High-bandwidth Digital Content Protection (HDCP).CEC is an HDMI feature designed that allows the user to command and control enabled devices. Based on the CENELEC standard protocol, it is a one-wire bidirectional serial bus that conducts remote control activities. Additionally the use of HDMI's HEC and ARC channels are two extra features. HDMI 1.4 added the HDMI Ethernet and Audio Return Channel (HEAC), which offers a high-speed bidirectional data communication link (HEC) as well as the ability to send audio data upstream to the source device (ARC).

ARC is an audio link that could be used to replace other cables between the TV and the A/V receiver or speaker system, with the TV generating or receiving the visual stream rather than the other equipment. Without ARC, the TV's audio output must be routed through a separate wire to the speaker system.HEC combines video, audio, and data streams into a single HDMI connection, as well as allowing IP-based applications and bidirectional Ethernet connectivity.

2.2.1.4 Digital to Analog Converter

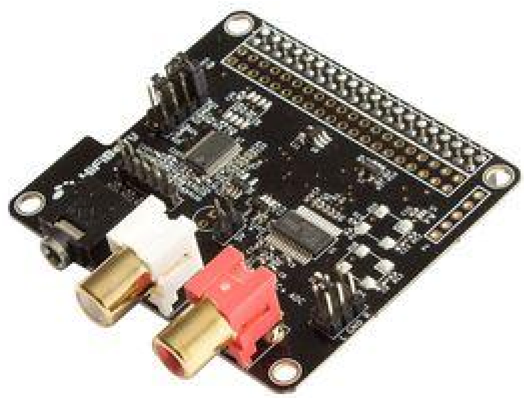
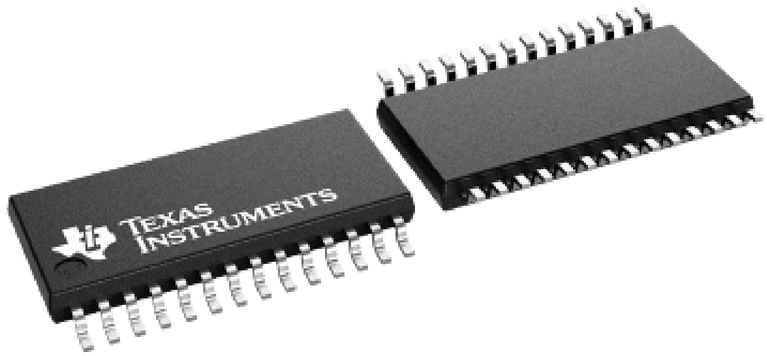


FIgure 6: DAC+ ADC Digital to Analogue Converter for Raspberry Pi

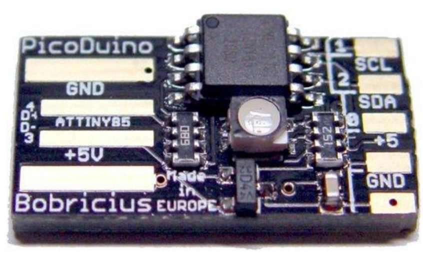
A digital to analog converter takes a digital signal and converts it into an analog audio signal. Any system that involves the playing of an audible sound sent from a computer requires a DAC. Without the use of a DAC the sound is merely just a collection of bits, a collection of 1’s and 0’s. The DAC is the middle man in regards to translating that code and turning it into a sound that is able to be heard. In regards to Smart Surveillance Hub the DAC is going to be the piece that receives a file from the Raspberry Pi and turns the signal into an actual alarm sound. The alarm sound will then be sent to the speakers from the DAC and the alarm will play. The DAC will also control the LED lights as well. The LEDs will always remain on and once the audio file is received from the Raspberry Pi it will trigger the LEDs causing them to change to a different color.

The specific needed DAC we need has to be able to be controlled by the Raspberry Pi using NodeJs which is the software selected for the project. NodeJs will communicate with the GPIO pins on the DAC and that is how the audio file will be sent over and able to be displayed on the speakers. The majority of current audio transmissions are recorded in digital form, such as MP3s and CDs, and must be transformed into an analog signal in order to be heard. Through the usage of Random access memory DACs, DACs are also employed in video sampling (RAMDACs). Due to the nonlinear response of cathode rays and the way the naked eye operates, video sampling had a distinct scale of work before the standard of LED TVs. The usage of RAMDACs in the design of a "gamma curve," which was supposed to produce the impression of equally distributed brightness increments over the display's full dynamic range, rendering hardcoding for each channel in a DAC obsolete.

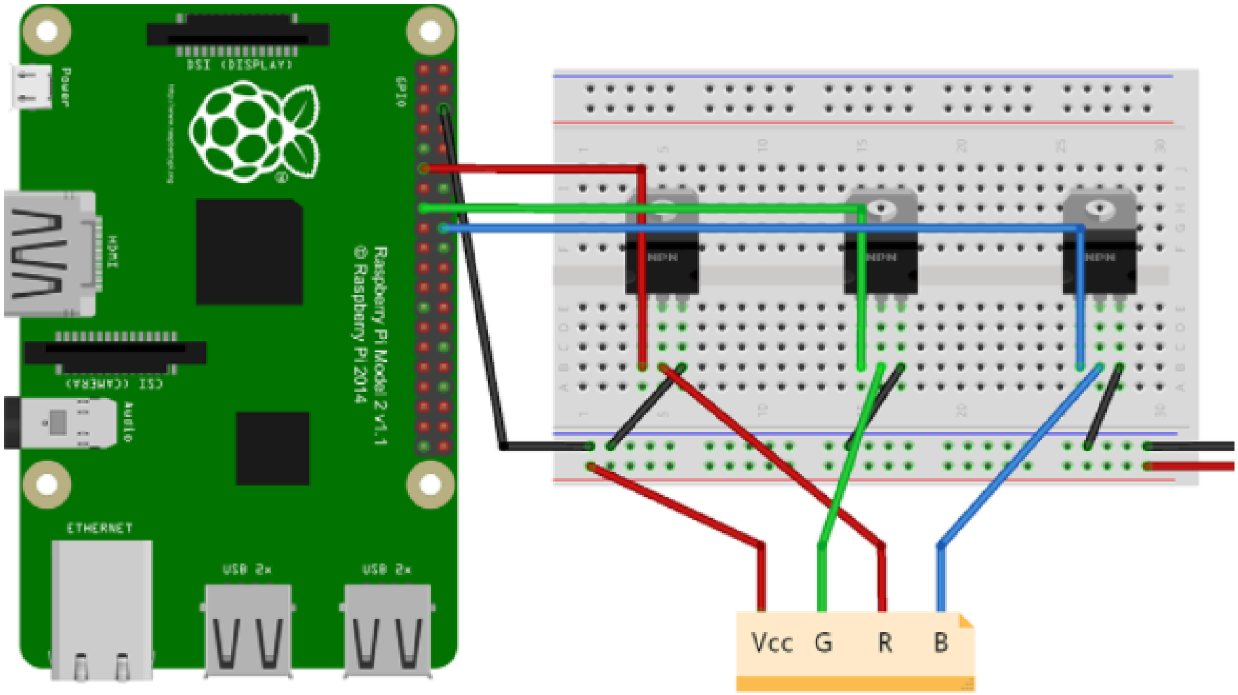


Currently our design for our DAC includes the DAC microprocessor PCM5122. It provides for CM Interface and Fixed Audio Processing With a two channel configuration,and a 32 bit resolution bus. This will meet our criteria for expanding our audio capabilities from our microcontroller.

2.2.1.5 LED RGB controller



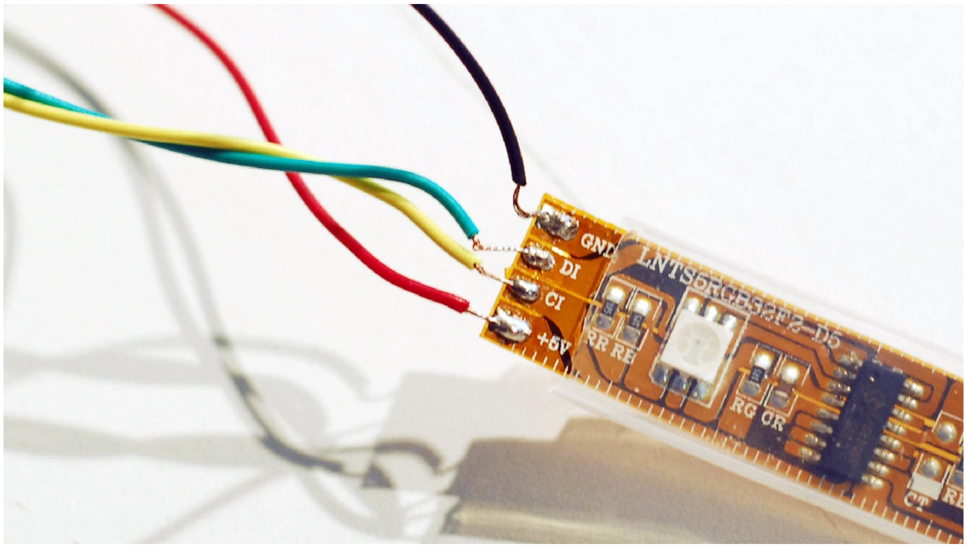
Another feature of the project is to have a visual representation of detection, to be more specific, we wanted to provide the user with a visual signal when someone was being detected. A simple way to do this would be LED alerts. The understanding of this idea was that when the device is idle power would still be applied to drive enough voltage for still green or blue LED light notification. Once our system detects someone on the premises this light would change to yellow or red. After some research I saw that this could be done using a programmable LED RGB controller or configuring the GPIO pins to create a simple ciuti for running an LED strip.



The idea of a simple RGB controller could consist of a LED light or strip, mosfet, load resistor (to keep the excess current from burning out the LED), and a power source. By configuring the LED with the mosfet gates we can control for what colors come on when we program or control the GPIO pins of the microcontroller. The connection for mosfet to LED should be as follows: Gate to GPIO, Source to LED, and Drain to ground.

2.2.1.5.1 LED strips

One of the most popular inventions in today's age is the LED strip, made extremely popular in 2017. LEDs strips can be seen in almost anyone's room ages 17-25. The versatility and options with these strips are endless. LED strips consist of a series of red, blue, and green light emitting diodes that in series with each other can be controlled to create color. When current flows through the diode a light is produced, when a different amount of light is produced in each of the 3 diodes different colors are generated. LEDs have many advantages over different light sources due to the low power consumption and the life of the diodes. LEDs require a direct current (DC) because a pulsating AC wave does not allow the diode to emit enough light consistently.



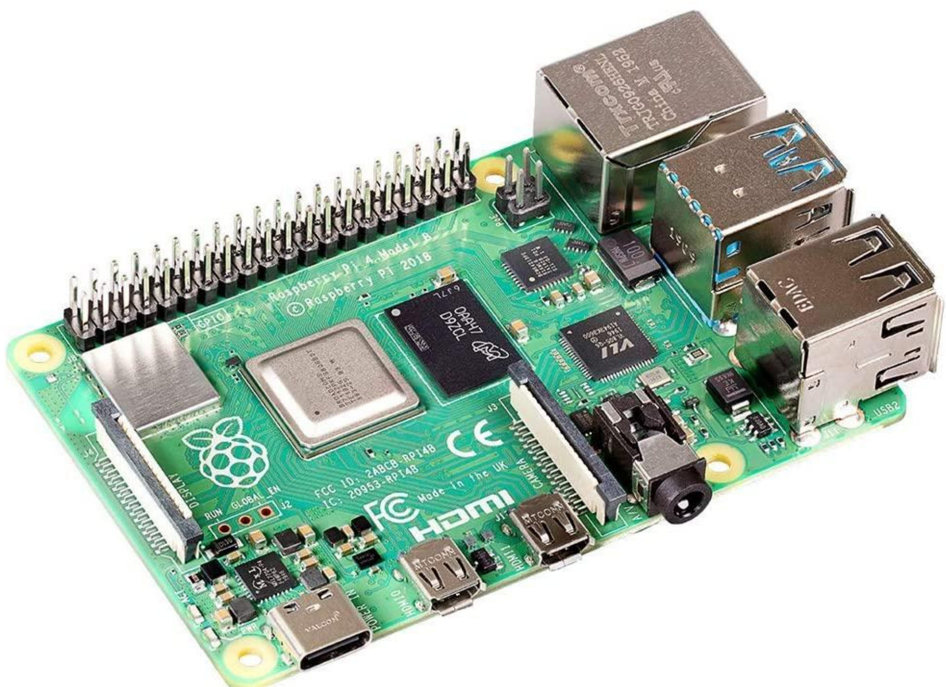
LED strips will be utilized in conjunction with a RGB controller in order to change the color of our Hub when an foreign person is recognized. These strips are vital to the Hub because if they are unable to provide a bright light the user may not notice the trigger of this alarm.

2.2.1.6 Single-Board Computers

Single-board computers (SBC's) are essentially microprocessors. They contain computer processors that perform the arithmetic and logic required to perform complex computing operations. We are most interested in Single-board computers that are low in cost but high in computing power. The majority of desktop computers run using Windows or macOS operating systems. However, most SBC's that are low in cost and heavily adopted in the scientific and engineering community generally operate using an open-source operating system. The SBC's we are most interested in are limited to a wide variety of Linux operating system distributions and aren't capable of running with Windows or macOS operating systems. Although Linux-based operating systems are not the most popular operating systems on the market, they are known to be potentially more powerful than Windows and macOS due to its open-source architecture. The IoT industry is well known for implementing and taking advantage of these low-cost microprocessors capable of providing the computation power required to carry out complex computing tasks.

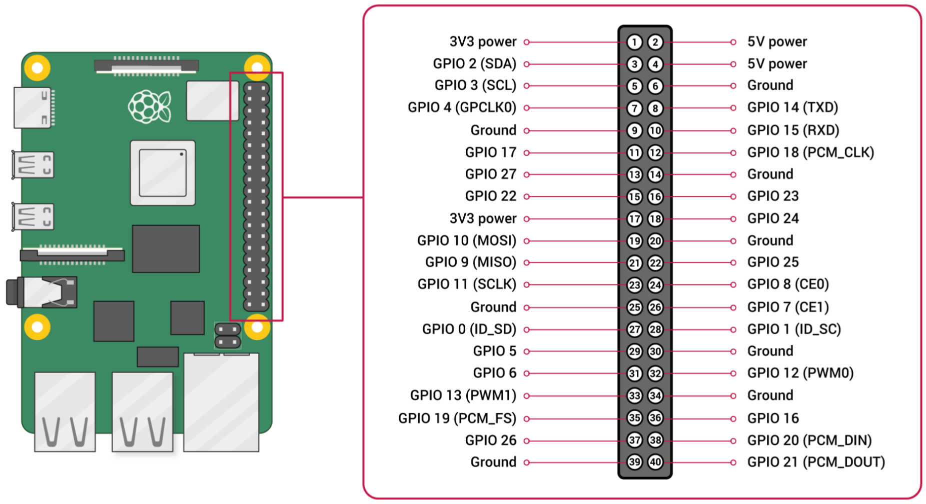
We've decided it'd be best to use a single-board computer since our project will process video and analyze it with deep learning in real-time. Since deep learning is a computation-intensive task, using a SBC gives us the required processing power to perform the many computer instructions necessary to operate human detection software. It also provides the critical performance needed to power the software responsible for running our desktop application to display the video camera's live footage. The SBC we use will act as our project's main component that communicates and controls the behavior of its external hardware. Most microprocessors are particularly good at running multiple processes simultaneously, especially ones with multicore central processing units. Since our project will be performing various tasks concurrently, we believe using a SBC with a multicore microprocessor is the best design choice for maximizing the performance of the application.

2.2.1.6.1 Raspberry Pi 4 Model B



Since 2012, Raspberry Pi's official release date, it has been one of the most commonly adopted SBC's used for building IoT applications. It's also widely used as a fully-functional desktop computer capable of running a Linux-based operating system. Raspberry Pi as a company offers a variety of different devices, all of them providing different specification tradeoffs. Due to its cost, physical size, and performance, we will conduct further research on the Raspberry Pi 4 Model B to act as the central hub of our project.

The Raspberry Pi 4 Model B uses a 64-bit architecture that utilizes a 1.5 GHz quad-core Cortex-A72 (ARM v8) processor. The SBC’s dynamic memory is available in 2GB, 4GB, or 8GB models. Since our project will require a large amount of data processing due to its human detection capabilities, we will use the 8GB model to ensure our project runs as fast as possible.



The Model B comes with a standard 40-pin GPIO (General-Purpose Input/Output) header that acts as an interface for the Raspberry Pi to control and communicate with external electronic circuits. The GPIO header can handle inputs and outputs, giving the Raspberry Pi the flexibility to add a wide variety of sensors and switches. The board also comes with plenty of available ports to connect external devices. These ports include but are not limited to two micro HDMI ports, a display port, and a camera port. The board uses the H.264 and H.265 protocols for processing videos. The H.264 protocol is capable of encoding and decoding video, while the H.265 protocol is only capable of decoding video. H.264 provides an efficient method for compressing the video and audio (encode) and sending the compressed data over some type of network. It can also accept the compressed video over a network and decompress it back to its original state (decode).

The Model B can run various operating systems. We’re choosing to use Raspberry Pi OS, the default Raspberry Pi operating system formerly known as Raspbian. Raspberry Pi OS is Debian-based; specifically, the version we're using is based on Debian version 11 (Bullseye). Since the operating system is designed specifically for the Raspberry Pi, we believe it’s the best design choice for maximizing performance and reliability. There is both a 32-bit version and a 64-bit version of Raspberry Pi OS. Raspberry Pi OS offers two 64-bit versions, Raspberry Pi OS Lite (doesn’t include a desktop version) and Raspberry Pi OS (consists of a desktop version). We will use Raspberry Pi OS (desktop version) because our project will consist of a GUI that controls the surveillance hub and displays the camera's views.

2.2.1.6.2 NVIDIA Jetson Nano

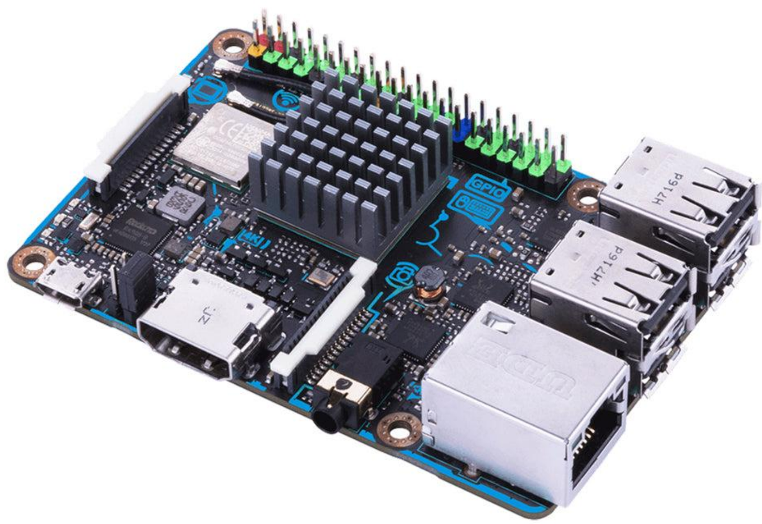
The NVIDIA Jetson Nano's official release date was back in March 2019. It has ever since increased in popularity and has been widely adopted by the scientific and engineering community. It is specifically designed for embedded systems applications, IoT-based applications, and training artificial intelligence models. The Jetson Nano is listed at an affordable price of $99 in the United States.

This powerful single-board computer uses a 1.42 GHz quad-core ARM Cortex-A57 processor that runs on a 64-bit architecture. It also contains an NVIDIA Maxwell architecture graphics processing unit (GPU) with 128 NVIDIA CUDA cores. It offers two different models that include dynamic memory composed of 2 GB or 4 GB of LPDDR4 that operates at 1.6 GHz and 25.6 GB/s.

The NVIDIA Jetson Nano contains the NVIDIA Jetpack SDK that supports plenty of artificial intelligence frameworks such as Tensorflow, Pytorch, Caffe, and Mxnet.

The board uses a microSD card that most commonly operates using Linux4Tegra, a Linux-based operating system based on Ubuntu 18.04. Although Linux4Tegra is the most widely used operating system, it is capable of running on a wide range of other Linux-based operating systems.

2.2.1.6.3 ASUS Tinker Board S



Launched in 2017 the single-board computer is designed to be a fully functional computer that offers class leading performance without leveraging hardware compatibility. The board is made for IoT.

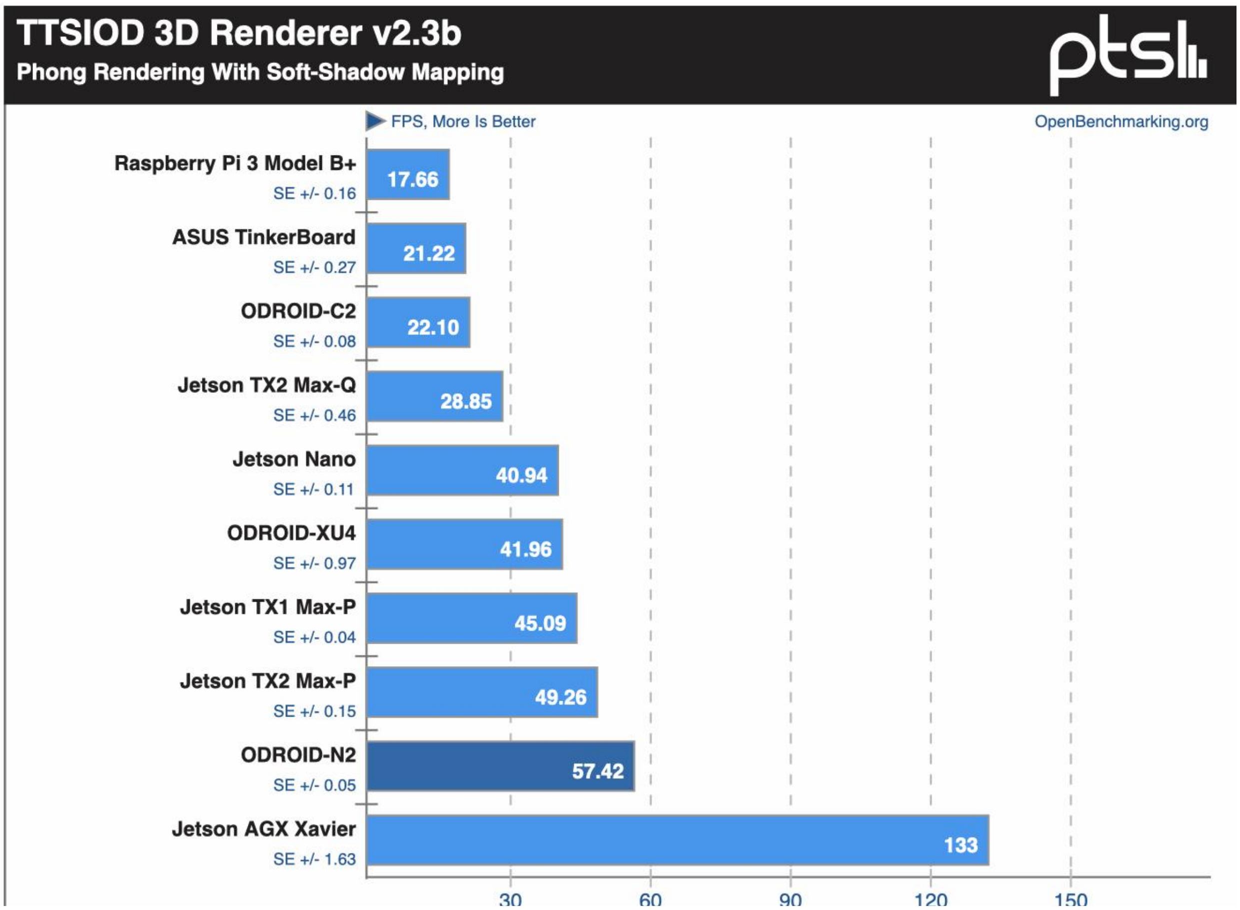
The modern quad-core ARM-based processor improves performance and provides flexibility to a variety of projects. The SD 3.0 allows significantly faster read and write speeds. The GPU allows for a high quality image processing and HD & UHD playback. In relation to the project this would allow the video playback to be smooth and quick on the hubs end.

The HD audio quality also differentiates itself from a lot of other SBC boards. The Tinker board features a HD codec that supports up to 24-bit/194kHz audio. The interface uses audio jacks which support audio output and audio input, without additional hardware.

This SBC includes popular applications such as IDLE/Python & Squeal/ Scratch. This board supports Android Operating System which allows media playback.

Rockchip RK3399
2X Cortex-A72 @ 2.0 Ghz
4X Cortex-A53 @1.5 Ghz
Mali T860 MP4 GPU @ 800 MHz

LPDDR4 2GB-4GB
16GB eMMC - Micro SD Slot
802.11ac Wifi & BT 5.0
12~19 V
Android 10 - Debian



2.2.2 Software Research

As per the software research, this section and following subsections will include information on potential paths and operations that will consist of different options to achieve the same end goals and initial specifications. In the software world, there are plenty of software stacks that operate differently internally but overall can achieve the same goals. This makes software a bit more difficult, because it puts us engineers in a tricky situation, giving us more options but making our choices more complex.

Researching the software before picking the first choice is the best practice because if we jump onto the first stack that seems to be good we might run into future issues such as the stack not having the capability to do things that we later need as features. That is why researching before developing is the most optimal path because developing and having to start over due to lack of research on the capabilities on the stack will ultimately waste time and resources of everyone involved and can create blocks within the development cycle.

The software requirements for the desired model of our project to work is that there will be a desktop application that is independent from any sort of web application in the user's interface. This will hold all the executable code so the user doesn't need to run any code environments or have any pre-existing software requirements, because the executable code should compile and run all within its own application environment. The application will be obtaining the RTSP stream from the user. Ideally we would like to automate this so the camera's stream automatically connects to the application, but every IP camera has a different unique RTSP URL stream that we cannot account for, so the user will be required to get the RTSP URL themselves and simply plug it into the client. The user will input this and send it to the business logic end of the code, which will handle all the streaming generation and use A.I. to evaluate the stream. Lastly, the software needs to be able to send signals to the Raspberry Pi-4's pin headers to output signals to the smart notifier.

After some research, we found two realistic options. These options consist of NodeJS and Tkinter. They have all capabilities to fill the needs of the software requirements previously mentioned.

2.2.2.1 NodeJS

NodeJS is a very powerful server side, cross platform and open source runtime software that has pre-existing and community created libraries and modules that can supply functionality to users giving people the power to create ground up applications. Traditionally NodeJS is used for web applications or websites, with the traditional Restful API with the client/server model. In this model, this allows users to see things on the front end and interact with things on a website and send requests to the server and can get responses which can be data from databases, such as a system to allow a user to log in or query and parse data from a database.

NodeJS as previously described has plenty of libraries, otherwise known as modules, that are used by the Node Package Manager (NPM) to allow users to use an extremely large variety of open-source software that can get the user capabilities and functionality to do things. We will look a bit in depth on what some of these modules are, what they can do and how they will offer us functionality to achieve our requirements and meet our deliverables.

NodeJS offers a bunch of modules, which act as libraries, that allow simplicity for using different libraries which are ported into these modules. These modules use javascript to utilize their functions, but give accessibility to things that other programming languages exclusively do. Modules are essentially a way to port other libraries from other languages.

One library we will need to communicate with the output pins on GPIO is called *onoff*. This module gives accessibility to all raspberry pi's, regardless of their generations. There are features within this module to essentially communicate with electronics to send on/off signals which will be used to toggle the MCU notifier to turn colors on the mood light or activate an audio cue.

There is a specification for A.I. for object detection, specifically human detection. An A.I. library needs to have the capability to have a model that can detect humans at the minimum, but for future stretch goals it'd be nice to be able to add and edit models to detect other objects in the IP stream too. A couple of options are either OpenCV or Tensorflow, as there are modules in Node that allow us to utilize their features within our NodeJS application.

Electron is a module that will allow us to port the application. Typically with NodeJS applications, the applications are localhosted within a web browser. Since Node acts as the software that can configure the backend server side management, ReactJS typically is the frontend software. Since React is the software stack's standard frontend framework, the default display is within a web browser. Electron makes it so that instead of an application existing only in a web browser, we can have a desktop application that runs your code as an executable file, but acts as a web application. Electron will be the way we wrap our application so it’s more desktop oriented and software portable across different platforms.

The server we are going to make to communicate with the client is an express back end. It's an open-source framework that will communicate with React using the Rest API, which is HTTP calls. The React client will send the URL needed for the RTSP stream and the server will do all the logic needed to run the stream from the camera. We will put the server on a port and the websocket we create with the RTSP stream will be on its own port, which the client will use to display the stream on the front end. The server will also use the node-rtsp-stream module that allows us to stream using any RTSP stream within the same network as the ip camera.

2.2.2.2 Tkinter

The alternative software stack is using Python. There are a ton of built in libraries and interfaces that will handle what we need to make a commercial desktop application to display the IP camera RTSP stream using A.I. detection and that will interface with the smart notifier. We will go into a bit of detail in how to achieve all the deliverables with python and how each function will use each respective library within python.

Tkiner is a built-in python interface within the python library that is used to create desktop applications. The name of the package is the “Tk interface”. It is otherwise known as a GUI toolkit. With Tkinter you can specify the size of the window and things that the application window can do. This will allow us to create an executable desktop application and is the alternative to the previously mentioned ElectronJS mentioned within the prior NodeJS section.

Within raspberry pi, the pi can run its micropython library that gives us accessibility to the GPIO pins and will allow us to toggle them for the smart notifier. In the micropython library, there is the machine library, which gives us access to the GPIO pins. This allows us to control and send signals to and from the GPIO header pins making it easy to interface with any PCB, or even a simple breadboard.

As far as the capability to stream an RTSP stream, python's OpenCV can take in an RTSP stream directly and also analyze it with A.I. to detect humans. Previously in NodeJS we needed two separate modules to stream and then to use an A.I. module to analyze it to detect humans. We can kill two birds with one stone using direct OpenCV to stream and detect objects.

2.3 Constraints and Standards

For now the main constraint for this project is going to be price. In order to develop a smart hub that is cheap and affordable it would be more beneficial to design and manufacture our own touch screen led screen. Since time is an issue and there are many other aspects of the project we are designing, we decided to buy a touch screen that currently works. If this project were to be marketed a cheaper touch screen led would have to be found in order to reduce the price as much as possible and make it as marketable as possible.

Another constraint is the need to figure out how every single camera works and creating a universal hub that works for every camera. Each camera has its own design and might not be universal to what we are creating. This is especially true since our target micro controller we will use is an 8GB Raspberry Pi4 B model. More software may need to be developed in order to become a universal hub for any camera.

Time can also come into play when developing this project, the hardware design and prototyping is going to take a bit of time and that is going to be needed to test the software. Many parts of this project need to be built in different pieces tested and combined after all have been tested. A Better illustration of the work needed for the project can be seen in the specification table (Table 1) and in the Project budget (Table 2).

2.3.1 Project Constraints

Throughout this subsection a variety of the constraints mentioned above will be further investigated in greater detail. Constraints limit how much of the product can be developed by us and how much of the project is going to be third party completed parts such as a LCD screen or LED lights. With more resources and less constraints these are things that could be developed by us and therefore delivering a more original product. Over this section the constraints that will be further explained will be time, budget, manufacturing, power, size, availability, environmental, political, social, and ethical.

2.3.1.1 Time

Time is the biggest limiting factor when it comes to this project. In roughly 6 months the product needs to be researched, developed, tested, and thoroughly documented. The research takes up a majority of the time and really limits the ability to thoroughly test the project. With more time different opportunities could be explored such as developing different parts that originally are being purchased, such as the LCD screen or the LEDs. Developing more of the project would allow for more money to be saved by not only us but by the consumer as well. In reality if there was more time, multiple prototypes could be developed and investigated which would result in a greater retail product.

2.3.1.2 Budget

Limiting the budget to around $600 causes the project to be as simple as possible. This budget only allows us to test one method of design. With a larger budget more designs could be explored and ultimately a better product could be developed. This budget allows us to buy a LCD screen as well as a Raspberry Pi, and leaves a little room for a few other important parts. With this budget we are aiming to create the cheapest product for our consumers. Certain parts are chosen because they are easily implemented into the project, when in reality we could have gone with a cheaper option which would have made the project more complex to create. It’s a simple trade off of time vs money and we chose saving time by spending more money.

2.3.1.3 Manufacturing

Access to large scale manufacturing is not available. Many of these companies have the ability to manufacture their own product. Since this product is a prototype it is not much of an issue with manufacture since we are purchasing each part individually. If this product was taken to market there would be serious issues relating to the purchase of each part. This also causes our retail price to be more expensive, with access to our own manufacturing method it would significantly lower the price and allow us to deliver a more market ready product.

2.3.1.4 Size

The casing of the prototype is 3D printed, so the size is limited to the dimensions of the 3D printer used. This poses a couple of issues because we are designing the project before we know the dimensions of the 3D printer. If the project we have designed turns out to be too large for the 3D printer we will have to find other options for creating the housing. On another hand if we design something that is large and heavy it will be a turn off for consumers because the industry has already stated a standard of size for similar products.

2.3.1.5 Availability

The availability of each team member has proven to be difficult. Everyone has different responsibilities and time availability. Multiple team members have jobs and full class schedules so even without designing a full project, their time is limited. This has limited us on the amount of time we are able to meet and design the project. If this was a product we were going to bring to market, we would be spending a lot more time together.

2.3.1.6 Environmental Constraints

The smart surveillance hub has only one environmental constraint and it relates to the power usage. The hub should properly manage power in an effective way to not draw more power than needed. If the power is monitored efficiently it will reduce the environmental impact of the hub as a whole. The hub will not cause any damage to the outside or inside of the home. Most of the time the Hub will be plugged into the wall in the central location of the home. The casing should allow for the hub to be protected and sealed from any outside tampering such as bugs or water. This will allow the hub to maintain its integrity and work for a long period of time.

2.3.1.7 Social Constraints

The smart surveillance hub is a social product just like most security systems on the market. If the system does not work as advertised, there will be terrible rep of the product and probably will not do as well. Cameras are the first defense of a person's home and family, if the hub does not alert the user it could result in harm to the home and lawsuits against the company. Every system that is built needs to be thoroughly tested to ensure that the clientele is able to trust this product with their life because in some cases they are. Security poses a constraint as well, if the product is not secure from network intruders then the system can do more harm than good. Ring recently went through this issue where their network was easily broken into and their cameras were used by a third party. This resulted in a social blowback and the media pushing back on their product tremendously. The clients need to know that when something goes wrong they are protected in their most vulnerable state, their home.

2.3.1.8 Political Constraints

The political constraints of this product relate to the network security of the hub. As long as the security of the hub is able to maintain itself on a private network and can protect itself in some way from foreign network intruders there should be no political constraints that limit our product.

2.3.1.9 Ethical Constraints

The ethics of designing a security hub system should be a main priority. The system and its integrity rely on the fact that it is reliable to protect the home. The system needs to be protected from cyber attacks and have measures in place in order to protect the privacy of the home. The only way someone could be hurt through the hub is if the hub was subject to a cyber attack and the intruder was able to view the cameras. Protection of the hub's network is crucial to avoid any issues.

2.3.2 Applicable Standards

This section will discuss a few standards in relation to the Smart Surveillance Hub. The standards will cover LEDs, software, power, PCB and soldering. Each standard affects the design of the hub.

2.3.2.1 Hardware/electrical standards

2.3.2.2 Software Standards

2.4 Design Overview, Technology, Architecture Content

This hardware diagram illustrates the connections between subsystems that form the single product. Every line directs the flow of data from the front to the endpoints of the product. There will be two main components after all said and done, the IP camera and the hub. The camera can be any off the shelf IP camera that streams an industry standard RTSP stream, while the hub will be the smart DVR that will be plug and play with the IP camera and will be housed in a singular unit that will be designed and 3d printed. Since the product is receiving information from the IP camera, this will be the start of where the data is taken.

The camera will be receiving information, this information being the camera stream. This streamed data will be sent over the network, which in our case and most cases will be over the wifi network, although it can be sent over a hardlined ethernet cable if the user desires. The IP camera and Smart Hub must be on the same network. If they are not on the same network, then the hub will not be able to connect. This is because the hub uses the camera’s RTSP stream which will only display on the same network. The network is the median between the camera and the hub in this architecture.

The hub consists of the main component, the Raspberry Pi 4. The Pi 4 is acting as the main computer for the hub, which will take and receive all the computational functionality from the camera and it will decide what to do with it. The Pi initially will receive the RTSP stream from the IP camera, and this will allow the user to be able to see the stream. This will also foremost ultimately be used in coherence with another software to determine human detection. The user will have an application that they can see the RTSP stream from the camera which will be a live stream of what the camera sees. This application can be seen on either a monitor or the hub’s built-in LCD screen, whichever the user chooses.

The firmware will analyze and dissect the stream using the A.I. software to determine if an object outside is a human or not; since this is a security system, we are only concerned about strangers and humans that are where they should not be. Things like cars or animals can be implemented but on this initial version of the project, we only care about humans. When the firmware of the application analyzes the data from the IP camera, it can then send this data to the MCU that will be the PCB notification feature. This MCU will be sent data from the Pi, through the firmware’s functionality to alert the user if there is a threat. There will be two main components of this notification PCB. The audio cue that alerts the user, and an idling mood light that will be certain colors depending on the ‘threat level’. Colors more aggressive like red will mean that there is a threat that is currently happening, or extremely recently just happened. Colors more calming such as blue or purple will mean that there is no threat and things have been, or are currently calm. All of this can be displayed below in Figure 2.

The software diagram illustrates the flow of execution between the data from the start to the endpoint. As previously mentioned, the start point is the moment that the IP camera records and streams the video feed, if this camera is not on then there is no flow of data or start point for the Smart Hub to function. In the software diagram, there are two components, the hardware and software.

The hardware is only there to fill in the gaps of this diagram, whereas we go into greater detail in this diagram on how and what software is truly needed to execute the overall operational goal. The camera will get the data and encode it to send over the network. There will most likely be some sort of SSH network protocol used to communicate between the Smart Hub and the camera. We need the video data encoded for fast and effective streaming capability. Without it being encoded, the data will be too big to send and it will have latency or resolution issues because the bandwidth or throughput will be suffering tremendously.

The industry standard of video encoding is using the H.264 codec which is typically a separate electronic device on a camera that handles it instead of the main processor, in order to alleviate the stress of a processor. Since we are encoding the data to send over the network, we will be decoding it on the hub’s endpoint after it is transferred over the network. This data will immediately be sent to the application, which will have the capability to use A.I. for human detection, to display the stream and to ping the notification PCB over the Raspberry Pi 4’s pin headers. We will also be planning to save the streamed data locally to a SD card. These are the core functionalities that will be on the first version of the product, with additional stretch goals desired depending on the time constraints.

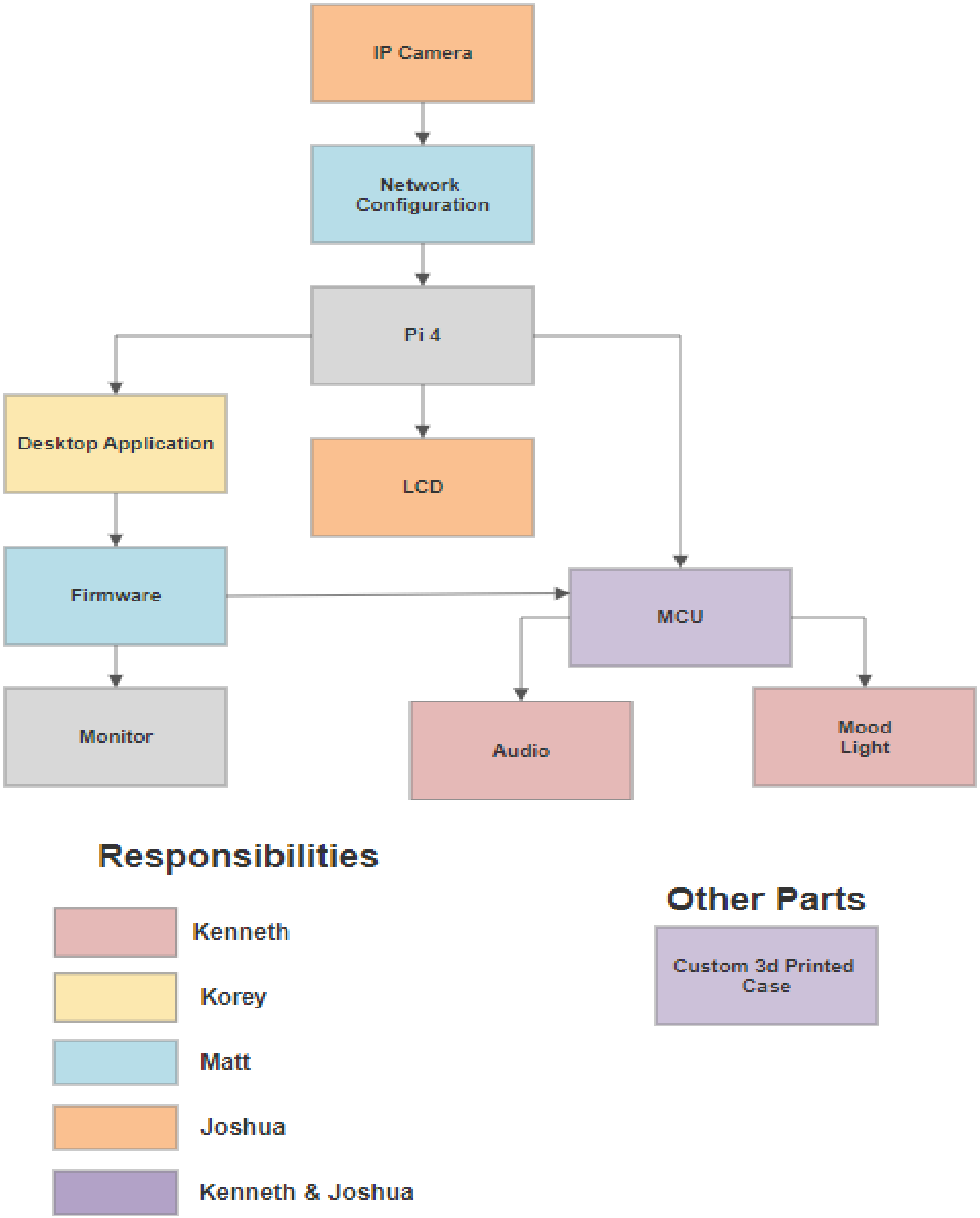


Figure 2:Hardware Block Diagram

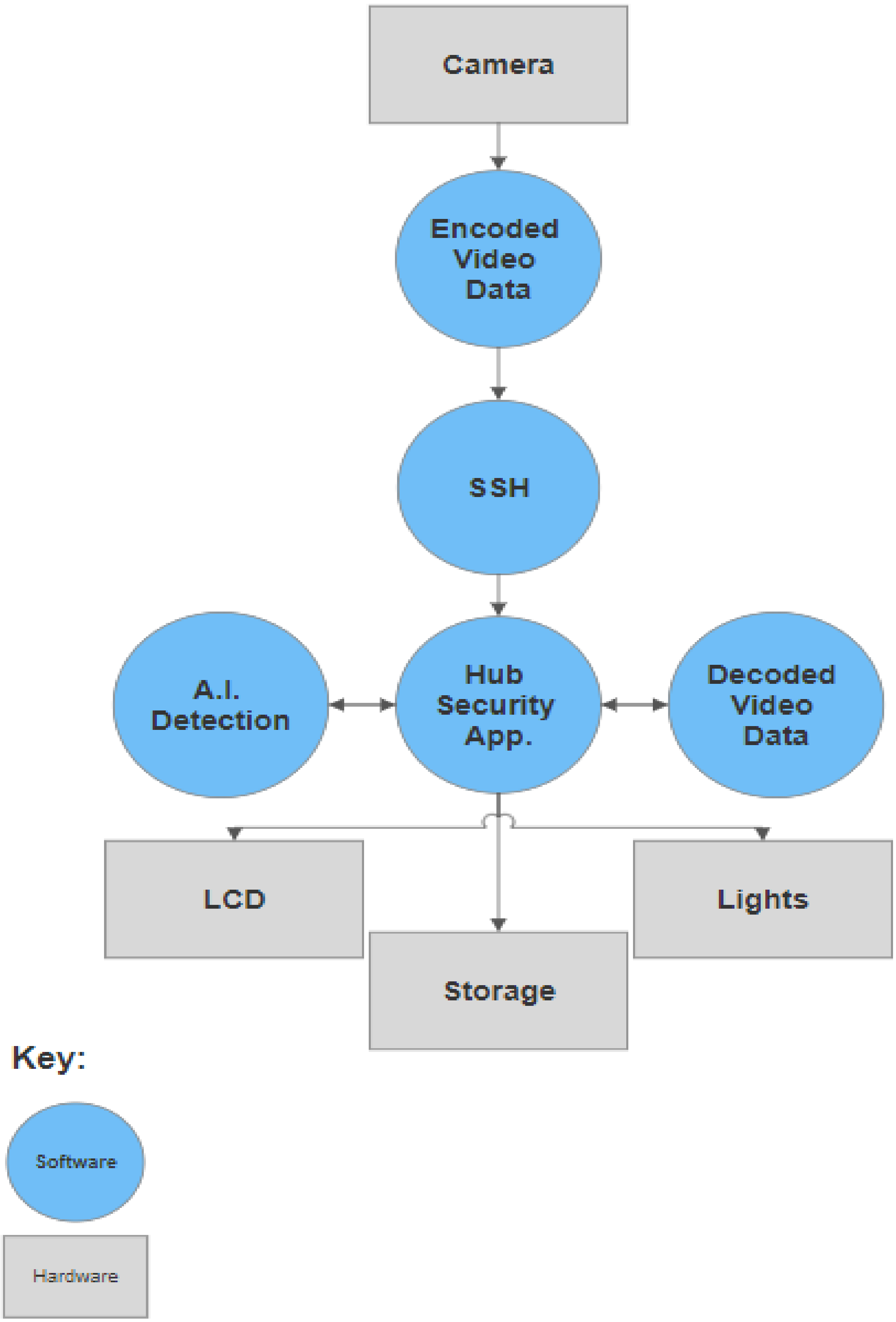


Figure 3:Software Block diagram

2.5 Explicit Design

2.5.2 Electrical Schematic Diagrams

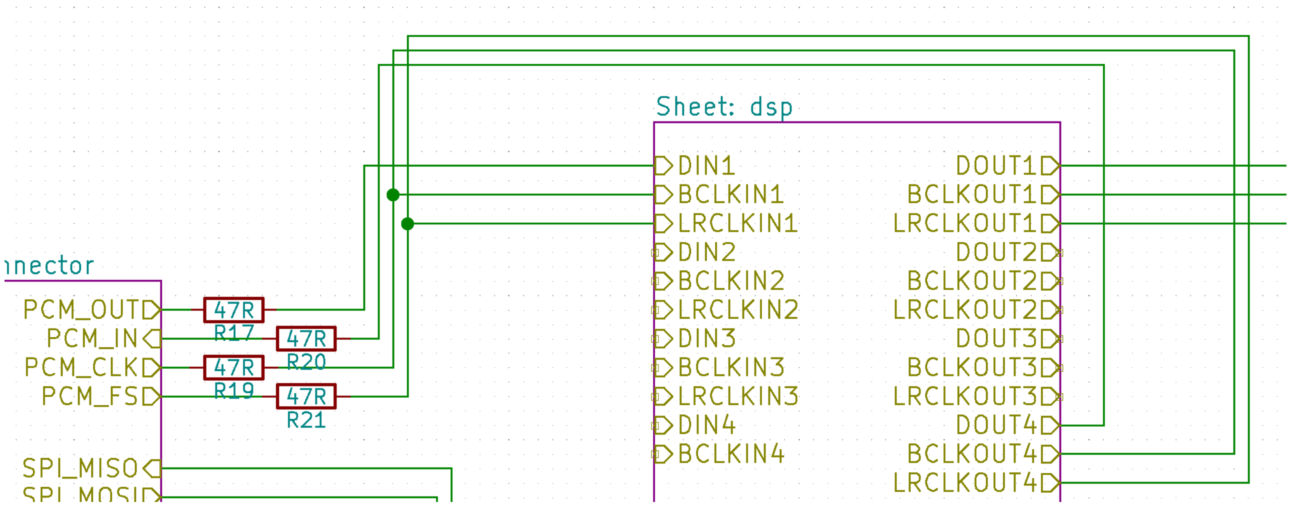


Figure :HifiBerry DAC schematic reference

2.5.3 Software Structure

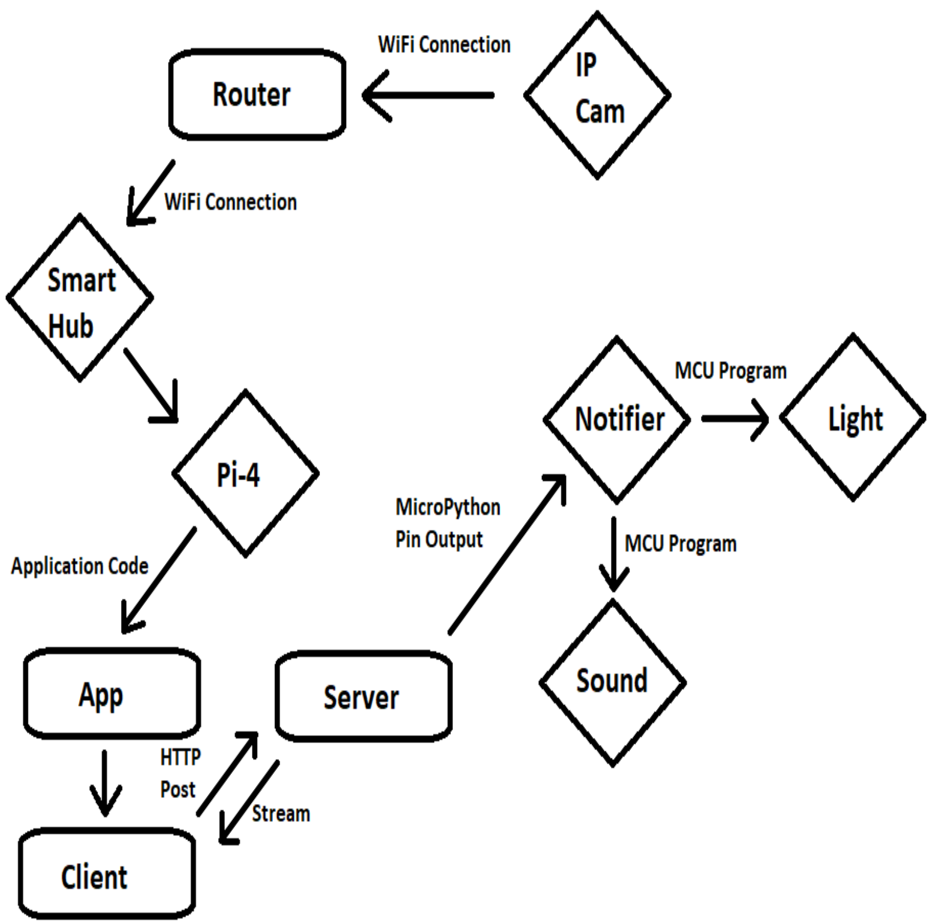


Figure 4:Software Structure

This diagram explicitly details the relation and usage of the software used to communicate between all the hardware and software. The arrows direct the flow of data using their respective network or communication protocols. There is only one arrow missing and that is the relation between the Smart Hub and Pi-4, this is because the Pi-4 is a subsection of the Smart Hub and the block for the Smart Hub is only there for clarity and not an actual part, but the main hub of multiple components.

The router will be the main traffic controller that communicates the Hub with the IP camera as discussed previously. The Pi-4's application will hold not only the client but also the server. They run in tangent and have very unique tasks. The client is what the user will be using to interact with whereas the server will run the stream, using A.I. detection and communicating with the notifier MCU to do certain tasks, which are conditional on what the A.I. will find out. Also, the client and server will be communicating with HTTP Post requests to send the RTSP stream from the user to create the websocket stream on the server, that will be displayed on the application's web browser using Electron.js.

2.6 Build and Prototype Plan

In this section, we will describe more in the subsections what our build plan, prototype, test and evaluation plan will consist of along with some additional information.

2.6.1 Build Plan

Starting from the top, the project has two of us CPE students working on the software side of things and two of the EE students working on the hardware side of things; therefore inherently there are two separate agile sprints going with different goals in mind. When developing the project, it's goals, the standards and requirements, this was a key factor in order to have independent sprints that can proceed on their respective workload that will not necessarily be given a block due to waiting on the other side of development.

When it comes to the build plan, the software is being developed at a steady pace, getting things working for the application. This is because the software development in the early stages of just the application can be engineered without any software being interfaced to it. The first thing we do is research what software stack can be used to achieve our desired outcome along with how realistic it is that we can achieve it with the current software. Realistically, almost any software stack can achieve anything in this day and age, just because of how far we have come with software, but our goal is to create the smart-hub app and to not develop an entire software stack from the ground up; which is why extensive research is important. If we decided on a stack and find out later that it can't handle what we require of it, then that can run into major issues and potentially will require us to scrap things and start fresh or we might not make our deadlines.

As per software development, the following goal after we find the stack we are confident about and desire to use, we then can start making a simple desktop application. This application's requirement is to just simply be able to run on a linux system, since our hub will be utilizing Raspbian, which is derived from linux, then the app must be able to run on that. When we have a base app that can open and run on the smart-hub, then we can start adding the baseline features to it.

The core baseline feature is being able to stream an IP camera within the application. This will be achieved by using the camera's RTSP url, which is different for everyone. An RTSP url string is camera and network specific, so it's not a static RTSP string. The RTSP consists of multiple things that is passed through one string, starting with the camera's username which can be different per IP camera, the password to access that camera, the IP address of the network the camera is connected to, the port that the camera is accessing on that network and lastly there is typically a generic string of some sort that is going to be different per camera model. Below is a generic example of an RTSP url:

rtsp://{username}:{password}@{ip address}:{port}/{camera model's string}

There are multiple ways for the user to find out their RTSP string, but this is required upon the user because we cannot predict or dynamically know their string, this information they would know since it has sensitive information like the camera's password and such. In retrospect, if the camera had no password then anyone on the network can access that camera via the respective port and hack into it, therefore this protocol is best for the user's safety and for simplicity on our end.

Since the application requires the user to know and put in their own RTSP url, there will be some sort of form input within the application when the camera is not being streamed or displayed on the app. This is the next step upon the application development, where we will generate a way to take in the user's input within the application for their RTSP url.

Upon receiving the RTSP from the user, we can work on the streaming process. Being able to stream the video is a must, because without it then the product is utterly useless.

Also without being able to stream it, then we can't operate on the stream with A.I. to utilize the PCB notifier device.

Down the pipeline of software once we build and develop the streaming capability, we can now finally use some object detection to check if humans are detected. After all, this is a security system, so detecting humans where they shouldn't be or simply detecting humans to let us know someone is approaching, is ultimately the goal of this device. Up to this point, an application and stream is working and so the A.I. software will be able to piggyback off the streamed data to then analyze it using A.I. modules. This development will happen at this stage and so there will be conditional code, whereas there is a very positive match for a human detection, then we can send that data to the next phase of the software pipeline for the hub to act appropriately.

At this point, we have all the software on the hub working for the application side of things and now we can experience a potential block, where now the application requires us to interface it with the hardware that will be developed.

Up to this point, we have a working application, a working stream and working A.I. to detect humans, the application is doing everything required of it on the hub's side of things. Now the firmware development would take place, where it will detect a human, show it on the hub's LCD or monitor attached to it and lastly send signals through the GPIO for the smart-notifier PCB board to do its thing. Within the application on the server side code, we will make GPIO pin header calls to send signals to the hardware to act appropriately depending on the use case.

As previously mentioned, if there is no hardware to interface with due to the PCB workflow creation, we cannot test the software on the desired PCB, but this doesn't ultimately mean we can't stop engineering. We can get a similar PCB and test code on that, so when we get to the point of this stage and if we are waiting on the PCB development, it's not a big worry because we can test on similar products and so this will not halt but rather give a similar product that will configure easily to the final PCB board.

In regards to the other side of the development and engineering, there will be that own research and creation process of the PCB. The first step is researching and making sure our options will fit our scope of what requires us to achieve our deliverables. The hardware needs to ultimately be able to communicate with the hub's Raspberry Pi 4's GPIO pins, or alternative pins for alternative methods. The PCB is going to be a DAC which will convert audio from the Pi to send the audio cue for the user to hear and know if there is a threat or someone detected. This DAC will be discussed in greater detail in its respective section within the paper. Also there will be another component that the GPIO pins will communicate with, that will handle the LED mood lighting displayed on the housing component of the hub.

2.6.2 Prototype

In this project, our goal as a group is developing a prototype as fast as possible so we have time to work out the kinks in the first version of it. Based on the build plan and previous section describing the workflow of the project, the prototype can be one of either two things, which we will discuss in a moment. First of all, the prototype needs to achieve a few things. These things consist of the pi having a desktop application that can stream and use object detection A.I. and this interfaces with some sort of PCB. Ideally, we would like the PCB to be the one we research, designed and created but depending on the workflow of the software side, we will most likely make our prototype using an off-the-shelf MCU that will be most similar to our PCB, so we can get the firmware working from the hub to the PCB.

Another thing is the housing for the product. The end goal is to create a housing for the hub that makes sense for the user and holds all the components so the final product is clean. For purpose and level of necessity, we will not be developing this 3d case until the very end; this is for a few reasons. For the first reason, the case is not something that will have any impact on the functionality, it is only meant to be done and used to have a better final product rather than some components grouped together within a random box. Another reason is that it is potentially not a valuable use of time, because the final product will probably change a few times on the hardware level of development and if we have a pre-constructed casing then we will ultimately spend more time constructing another case to fit the new(er) versions of our product.

So with all of this being said and done, ultimately the prototype will consist of the core functionality to at least interact with the hub and a DAC. This will be the biggest challenge and getting that working will be a tall task to overcome, because there are a lot of requirements leading up to the prototype. On the second phase of the prototype assuming we have the DAC working, we will get the lighting and other components on board with the software, but these are lower severity; although they are deliverables, they are just more simple to implement and will most likely have less things to overcome to make this work. If we want the lighting to be implemented in the first prototype, we can always just use a breadboard to test lighting capabilities form the GPIO but this is the easiest thing to do with an MCU so that’s why our time is initially going into the more difficult stuff.

2.8 Facilities and Equipment

The scope and goal of this project is to create a product that can simply replace current solutions and interface with ‘off shelf’ IP cameras. This entails creating an at-home security system, or at least a security system anywhere that there is an internet network. As per the facilities, testing and development of this project will be done in the same type of environment that the goals entail.

As far as the development and end-goal environment for our product, we will be developing the project and hardware in the comfort of our individual abodes. This is good because ultimately this is the target environment for the end-product with all of our deliverables, will be that this system resides in a residential or commercial space. This requires internet access for both the camera and the smart-hub, because that’s the communication median between the two.

As previously mentioned, the project consists of creating a smart-hub that is an all-in-one system that interfaces with the camera. External equipment that is required, but not engineered or created by us is an off-the shelf IP camera. Since the goal is that our product can interface with one of these current production cameras, we want to get an ideal simple camera that offers all the capability we desire. Creating our own camera for this project would be useless and defeat the purpose of the product, which mission is to be a plug-and-play hardware interface with software to display and use any IP camera; therefore it’s ideal that we simply spend our time in researching, developing and engineering the hub rather than any sort of camera.

The last requirement is that there must be internet. This can be achievable by a simple router or wifi. Each IP camera is different, whereas ours that we bought utilizes an ethernet cable to connect to the network and not a wireless internet connection. Since our current camera we are using to produce requires an ethernet cable, then a router is required for development purposes to get the camera on the internet.

2.9 Testing

Testing our application is a very critical step for delivering a product that the user’s are going to be satisfied with. By conducting extensive testing, we will minimize the amount of possible bugs that could be detrimental to our project’s work flow. Our testing procedures will include software and hardware. The software will be tested to ensure that each important aspect of our project performs and behaves as stated in the requirements. Our project’s end-to-end testing will be conducted by analyzing each of the project's required tasks and laying out detailed testing plans to verify it's behaving accordingly. The software will also be tested using the Jest testing framework that will carry out our project's unit testing and integration testing. Testing our projects functionality will give us the confidence that our project is ready for production and these steps are equally as important as any of the other steps that go into designing our project.

Unit testing does exactly as it sounds, they test individual pieces of the codebase, also known as units, to ensure each component is working properly. When conducting this type of testing, the unit being tested should not be connected to any other units of the application that could have a strange effect on the output. Handling test cases by using this technique makes it easier to diagnose bugs and correct them without affecting other parts of the system. The key advantage of running unit tests is they are very good at isolating bugs since the components are tested separately and the code being tested is relatively small.

Integration testing is generally conducted after all unit testing is successful and before conducting any end-to-end application tests. The main point of integration testing is to verify that all of the working modules and components behave as intended when they're interacting with the other modules and components within the system.

End-to-End testing will be used for proving the functionality of our software meets the requirements stated throughout this document. This includes any major requirement that we promise to the user will work as intended. When conducting end-to-end application testing it is important to think of cases that aren't likely to occur often but could cause serious damage to the system if not handled correctly. Before conducting each test, there will be a desired result we are looking for and if the result differs from this, we know the code has a bug somewhere. If the tests pass this doesn't mean our code is bug-free, but it does mean our application works for the test cases we passed to it. That's why it's important to run many tests, and to think of certain cases that could potentially cause problems for the system.

Oftentimes projects that get rushed into production lack thorough testing. Due to a demand on shipping the product quickly this step gets skipped too often and because of this can end up with many bugs and unhappy users. Many times a project will go into production because the creators believe the product is in a working state because they only run a few tests that pass and figure their product is ready to deploy. There are many important tasks that our project is responsible for delivering. If any of these components fail due to lack of testing it can be detrimental to the product and any of the user's who are using it.

2.9.1 Hardware Testing

Purpose:
Hardware testing could consist of taking pre-made pre-existing components and testing them as you would your ideal design. This can be done using a constructed prototype or using other finished products to test your idea. The primary goal when testing hardware is to ensure the certain constraints are being made. Within an electrical system components can be sensitive to changes such as an increase in voltage or changes in frequency so it is pertinent to test for certain causes of potential failure.

Supplies:
- IP camera
- Raspberry Pi
- Prototype of or PCB
- All other hardware that completes the system schematic software
- Multimeter or

Procedure:
1. Create a testing environment (e.g., measurement hardware, test software, cabling, fixtures, etc.)
2. Place part into the condition needed for the measurement (apply pressure, voltage, temperature, etc.)
3. Take some measurements
4. Put those measurements through one or more pass/fail criteria
5. Record the results as either summary data or verbose raw plus summary data

Results: Test has not yet been conducted.

2.9.1.2 Unit Testing

2.9.1.3 Integration Testing

2.9.1.4 Initial Camera Configuration Testing

Purpose:
When a user uses our product for the first time, some initial configuration must be done to connect their IP camera to the application properly. This part of the testing will ensure that the information the user inputs will correctly set up the camera dynamically. The user will be brought to an initial landing page that asks them to input their camera's IP

address, domain name, and password for the application to gain access to the camera's footage via RTSP protocol.

Supplies:
- IP camera
- Raspberry Pi

Preparation:
- Create a landing page consisting of a form for users to input their credentials to access the camera's input stream.
- Implement dynamic variables in the application's code that will be concatenated to create a dynamic RTSP stream configuration.
- Connect the Raspberry Pi and the IP camera to the same network.

Procedure:

1. Once a user enters their credentials into the Raspberry Pi, the information will be saved to the device's operating system. The first step in testing this is to make sure any existing environment variables previously used to save the user's credentials are erased.
2. Power on the Raspberry Pi and the IP camera. Once the device is powered on, a script will automatically boot the application.
3. The system will not know the user's credentials, so it will bring them to a landing page asking them for the necessary configuration details.
4. Enter the camera's IP address, domain name, and password.
5. Ensure the user's camera is appropriately set up by viewing its RTSP stream.
6. Reboot the device and make sure its credentials were saved correctly and the user is no longer brought to the landing page.

Results: Test has not yet been conducted.

2.9.1.5 Testing the Deep Learning Model

Purpose:
The purpose of testing this part of the software is to ensure that the system detects humans without any deficiencies whenever someone is approaching the camera. Ideally, the camera should detect a person from a range of about twenty to thirty feet. However, it is critical that the system absolutely makes the correct detection when a person gets closer to a range of about five to ten feet. The model we are using for

human detection makes predictions using an accuracy-based approach. We'll set up our application to trigger when the model's prediction accuracy rate is 80% or higher. Therefore the further away the human is from the camera, the less accurate the model's prediction may be. But when the user gets closer to the camera, the prediction will become more stable. So to ensure the system doesn't make any false predictions, the model's accuracy rate cannot be lower than 80% before making an assumption. This part of testing will also be set up to ensure it detects humans when the sun is out and when it is dark outside. It will also test for humans wearing different types of clothing, from hats to hooded sweatshirts, and humans with facial hair and humans without facial hair.

Supplies:

- IP camera
- Raspberry Pi
- Linux terminal
- Measuring tape
- Hat
- Hooded sweatshirt
- Four or more humans varying in hairstyles and facial hair

Preparation:

- Design the application to trigger an alert when it makes a prediction of 80% or greater.
- Gather each team member to participate in testing.
- Conduct testing when it's very bright outside and very dark outside.

Procedure:

1. Place the camera at the front door of one of the team member's house or apartment during a bright sunny day.
2. Boot the system and wait for it to initialize.
3. Have one of the team members walk up to the front door. Wait for the system to trigger it and detect a human.
4. Record the prediction's accuracy and the human's distance from the camera when the event was triggered.
5. Repeat steps 1-4 at night when the sun goes down.
6. Repeat steps 1-5 for each member of the group.

Results: Test has not yet been conducted.

2.9.1.6 Testing Audio and LED Alerts

Purpose:
It is very important that our deep learning model passes all of its tests and the model has been proven to be successful at detecting human presence. After the model has been proven, the LED and the audio component must be put through extensive testing. It is important that the user is alerted with both visual alerts and audio alerts. When a human appears on their property. Although testing the model is important, it is equally important that the sound hub triggers an audio alert and the LED's produce light to alert the user.

Supplies:
- IP camera
- Raspberry Pi
- Linux terminal
- Sound hub
- LED's
- Two humans

Preparation:
- Design the application to trigger audio alerts and glowing LED's when a human is detected by the application.
- Design a testing program that generates simulated human detection instances replicating the deep learning model. The program will consist of a loop that produces fake deep learning predictions that are supposed to cause the applications alerts to be generated. The loop will run for one minute on and off increments. For the minute the loop is generating outputs, the alerts should be triggering and for the minute the loop is not active, the alerts should not be triggered since it's simulating no human presence.
- Position the camera at such an angle that it can pick up humans from a wide range of spectrum.

Procedure:
1. Power on the Raspberry Pi and the IP camera. Once the device is powered on, a script will automatically boot the application.
2. Have one person stand outside from a distance far enough away that the camera cannot make any detections. The other person thats sitting inside next to the hub will be on a phone call with the person outside to help instruct them on what movements they need to make to trigger the alarm.

3. Once the person inside verifies the application is in a ready state, they will instruct the person outside to start walking up the front door where the camera is positioned.
4. The person that is sitting inside the house will observe the LCD display to watch how close the person gets to the camera when the human gets detected and the alerts begin to fire off.
5. The person located inside is still connected to the person located on the outside through a phone call. Once the model has made the proper human detections, the person located inside will direct the person to slowly back away from the front door until the person is out of the camera's range. The person located inside will continue observing the LCD display to verify that once the person exits the premise, the application no longer triggers alerts and the user of the system will then know the person has exited the property.
6. After steps 1-5 are complete, it's time to pass the application some fake data that replicates the applications desired deep learning predictions.
7. Using the linux terminal, run the program that loops in one minute on and off iterations. One minute the loop will produce simulated human detections, and the other minute it will do nothing.
8. Observe the sound hub and LED's for the first minute verifying the alerts are triggered are they are supposed to. The other minute, observe that the sound hub and LED's aren't triggering any alerts.

Results: Test has not yet been conducted.

2.9.1.7 Handling Undesired Input Testing

Purpose:
Our application must be able to handle incorrect input data without bringing down the system. One form of bad input data that must be handled correctly is the case when the user inputs their RTSP stream credentials. It is important that if the system is given incorrect data while trying to connect the camera to the system, that the system doesn't store the incorrect values and the system will not crash. In this scenario, the user should be notified that the input in incorrect and they should be brought back to the landing page where they can input the data correctly. Another form of bad data the system may encounter is a model detection that differs from a human. The deep learning model we are using for the application is a pre-trained model called TensorFlow Coco SSD. The TensoFlow model is capable of detecting many objects that it may encounter that are not humans. Testing this part of the application will ensure our application will not trigger and alert if it encounters another object like a bike or an animal.

Supplies:

- IP camera
- Raspberry Pi
- Linux terminal
- Sound hub
- LED's
- One human

Preparation:

- Design the application to redirect the user back to the landing page where they enter their camera's RTSP credentials if it is given incorrect input values.
- Design a testing program that generates simulated detections of instances that differ from humans that the deep learning model is capable of producing. The program will consist of a loop that produces fake deep learning predictions that produce detection of different possible objects. We will use a different type of object in each iteration of the loop to be sure there aren't any scenarios when an object different than a human might set off an alert. The loop will iterate through each type of possible object detection provided by the TensorFlow Coco SSD model. After each type of object is passed to the application the person analyzing the test will verify that the system did not trigger any alerts for any of the incorrect data inputs.

Procedure:

1. Remove any previously stored RTSP credentials that were saved as environment variables on the Raspberry Pi operating system.
2. Power on the Raspberry Pi and the IP camera. Once the device is powered on, a script will automatically boot the application. Since the environment variables were erased, the user will be brought to the landing page where it will ask for their RTSP credentials.
3. The user producing the test will purposely input incorrect RTSP values and attempt to generate a connection between the camera and the application.
4. The user conducting the test will verify that the program redirects them to the landing page to reenter the RTSP credentials. They will open the file stored on the Raspberry Pi's operating system to verify that it didn't store the previous incorrect input as an environment variable since the data is capable of making a connection to the application.
5. To test the application for undesired model detections, the user will run the test program that's designed for passing the application generated model detection of all the different types of objects the model is capable of detecting.

6. To verify the application is behaving as intended, the user who is testing the application will evaluate the behavior of the hub's notification system verifying the hub does not trigger any LED or sound alerts when encountering an object that differs from a human.

Results: Test has not yet been conducted.

2.9.2 Complete System Testing

In this section we will lay out a variety of tests in order to test the range of cameras the system will work with, the durability of the housing, and the response time of the system. Each one of these is critical in bringing a complete product to market. The range of cameras the system can work with is vital because the point of the hub is to be able to purchase any IP camera and hook it up to the system. The durability of the hub is important because the system is going to be running constantly so the electronic parts are going to heat up. Finally the response time is important, if the camera has a significant delay then the system does not work as intended.

2.9.2.1 Flexibility Testing

Purpose:

The purpose of testing the system flexibility is to make sure the system works for a variety of cameras. The variety of cameras the system can connect to will provide information as to if the system is viable to the market. Connecting to 3 or more cameras will prove that the system works as intended and will most likely work for each IP camera the consumer purchases to work with the system. If more than 1 different type of camera can be connected to the system at the same time that will prove that the system will do well in the market.

Supplies:

- 3 Different IP Cameras
- Raspberry Pi
- Linux Terminal
- LCD screen

Preparation:

- Select 3 different IP cameras from a third party distributor like Amazon.
- Set up each camera to make sure each is working based on their user manual. Go into the network and make sure each camera is connected to wifi and is streaming a steady video.

- Power the raspberry pi and make sure the system is running properly.

Procedure:

1. After the first IP camera is on the same network as the Raspberry Pi, the user will look on the terminal for the camera's RTSP credentials.
2. Enter in the camera's username and password.
3. Check on the LCD screen to verify the camera is displayed and working properly.
4. Look for the 2nd different IP camera on the same network as the Raspberry Pi, on the terminal look for the camera's RTSP credentials.
5. Enter the camera's username and password.
6. Check the LCD screen to see if the camera is located next to the other camera and both are working simultaneously.
7. Finally look for the 3rd different IP camera on the same network as the Raspberry Pi, on the terminal look for the camera's RTSP credentials.
8. Enter the camera's username and password.
9. Check on the LCD screen to see if the 3rd camera is displayed and working properly.

Results: Test has not yet been conducted.

2.9.2.2 Durability Testing

Purpose:
Testing the durability of the system is critical, if the system is prone to overheating or there are power supply issues the system will not run for a long period of time. In order to test the efficiency of the system. The thermal output of the Raspberry Pi is crucial to understand the longevity of the hub.

Supplies:

- 1-3 IP cameras
- Raspberry Pi
- Prototype of or PCB
- Linux Terminal
- LCD Screen

Preparation:

- Turn on the system and assure the system is running the AI facial recognition software continuously.
- To start connect 1 of the IP cameras to the system and assure that a feed is present on the LCD screen.

Procedure:

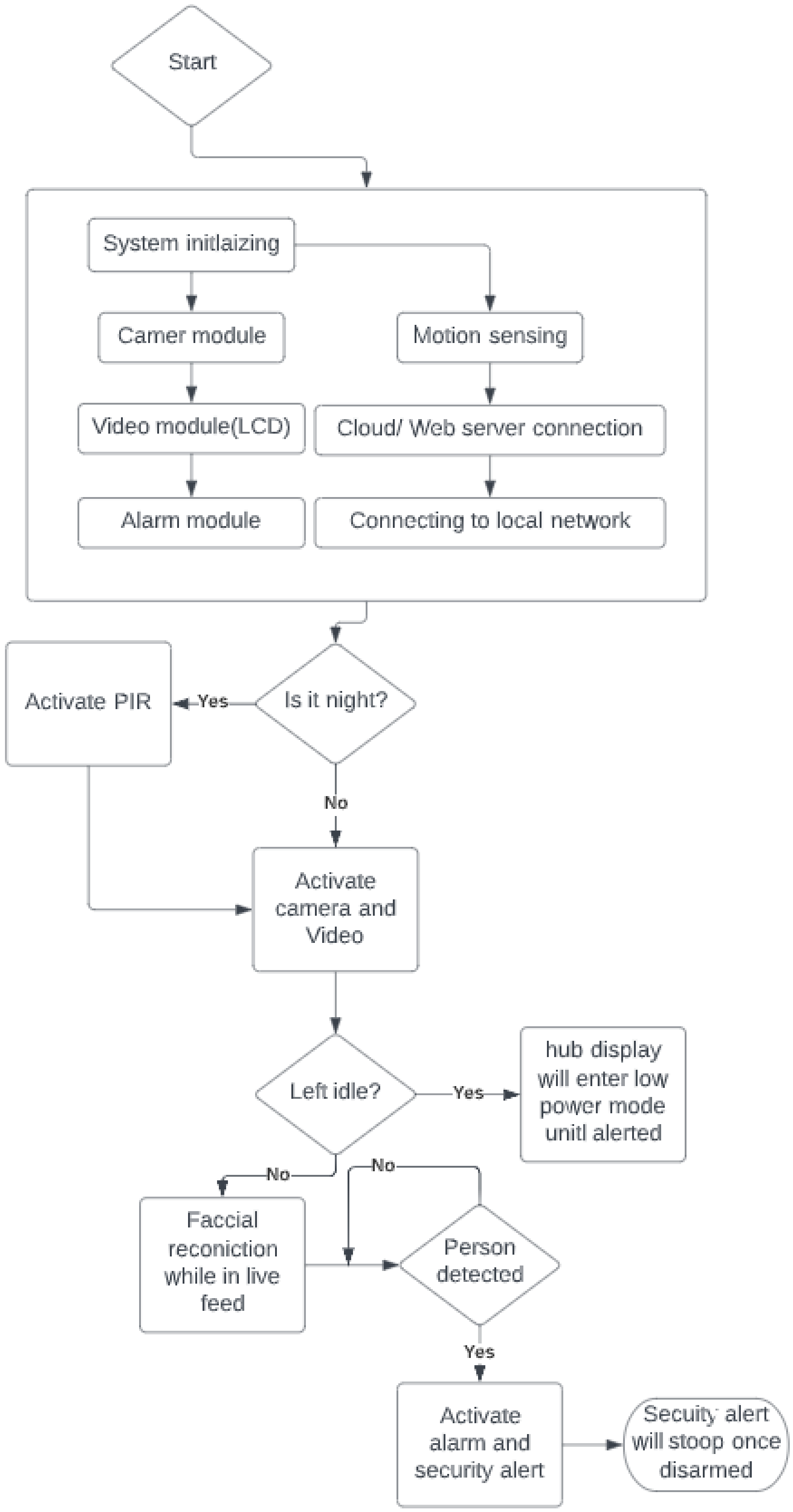
1. Open the pi@raspberrypi terminal and run the command prompt “vcgencmd measure_temp” this will result in a temperature reading of the GPU.
2. Record the temperature of the GPU and record the time.
3. Next on the same prompt window run the command “cat / sys/ class / thermal / tehrmal1_zone0 / temp.”
4. Record the temperature of the CPU and divide the number by 1000 and the reading will result in a temperature in Celsius.
5. These numbers need to be recorded every hour for at least 24 hours in order to receive a complete set of data.
6. The data then needs to be compared to the acceptable range of the Pi.
7. Next add more cameras and record that data and compare again to the acceptable range.

Results: Test has not yet been conducted.

2.9.2.3 Response Time Testing

Purpose:
Response time is important in every application. Regarding security, the response time is critical. If the response time of the camera is significantly delayed it could cause issues for the consumer and ultimately rule the product useless. A great response time will be anything

2.10 Project Operation
2.10.1 Operation flow chart


Start
System initlaizing
Camer module
Motion sensing
Video module(LCD)
Cloud/ Web server connection
Alarm module
Connecting to local network
Activate PIR
Yes
Is it night?
No
Activate camera and Video
Left idle?
Yes
hub display will enter low power mode unitl alerted
No
No
Faccial reconiction while in live feed
Person detected
Yes
Activate alarm and security alert
Secuity alert will stoop once disarmed

2.11 Stretch Goals

- Cloud storage capability

  Storing data that was recorded from the cameras in the hub and transferring all the recordings to the cloud would be a great addition to the system and allow for a greater product. Time is the main constraint when dealing with this task.
- Mobile application

  Developing a mobile application that will be able to stream current camera footage. Also being able to see the recordings on the cloud storage. This would be able to access outside of the local network of the IP camera and hub.
- Backup battery

  A backup battery in the hub would allow the hub to be more mobile for the customer to carry around and still be able to have access from a non-stationary location.
- Developing our own IP camera

  Creating our own IP camera with off the shelf devices is the ultimate stretch goal. It would allow the product to become a complete system with recording and storing as well as smart notification services. Time and complexity of this task is what makes this a stretch goal. If the hub is completed in a timely manner then this will be the first thing we try to develop.
- Adding increased functionality to control other types of IP Cameras

  Cameras like PTZ cameras come with extra features to pan tilt and control zoom. Allowing access to these features through our hub would give the user more control over how they would like to use their device.
- Ability to cast received data

  As a higher level feature we would also like to include the ability to cast data received from the camera to give users the option to see video without a hardwired cable connected to their display.

3.0 Related Products

Security companies have been on the market for many decades, but we believe there are no products similar to ours in terms of simplicity and flexibility. In this section, we will look at multiple products similar to ours to better understand the market. Our research on different products will focus on the product's physical appearance, usability, flexibility, and overall cost. Each security company that is mentioned has similar qualities to what we are offering. These companies have been doing similar things for the past 10-15 years but often lack the flexibility we offer. Throughout this section we will also do a deep analysis on other products’ features such as video-streaming quality, wifi connectivity, installation techniques, and other essential features that may cause the user to like or dislike a product. By comparing other products' pros and cons, we’ll be able to gauge which features the current products lack and determine which features we can add to our project to make ours more powerful and unique.

3.1 Vivint

Vivint is a well established security company that provides many useful security features. They have a highly reputable product that has a broad range of services ranging from front door cameras to a car theft prevention product called Car Guard. Their cameras are what we are most interested in. They offer doorbell cameras, indoor cameras, and outdoor cameras. The doorbell camera pro and outdoor camera pro are the products that relate the most to our project since their main focus is surveilling the property of a user’s home. Both of these cameras are installed using wires ran through the housing complex. These cameras are wired for power but not for connection which makes them an interesting company to talk about. Their cameras are very flexible, they offer cloud recording at an industry leading quality along. They also use WiFi, have night vision recording, motion detector sensors, two-way audio, and they’re weatherproof. The camera footage is stored in the cloud that can be accessed anytime from the user's smartphone app or from the user’s hub located inside their home. All of their products are highly reviewed. However, to get a straightforward answer on how much the product costs, a sales representative has to come to your home and do an inspection.

An important comparison that we are going to look at is their smart hub. The smart hub is a central monitoring control that is hung on the wall. This hub is wired for power and has an array of features. This hub that Vivint creates is basically the brains of their system. Roughly the size of an ipad it is a stationary hub that lacks on a few things, it is unable to move and unable to be unplugged or the system will set off an alarm. The Hub also allows for video surveillance but only from one location in the home. The hub they offer has been improved constantly every 2 years and is software updated around every 3 months. The hub is very user friendly and allows for virtually anyone to use but requires a passcode to login and view cameras. The thing that elevates this hub is it has two 4G LTE chips in the hub which allows it to maintain a steady connection even when the wifi shuts off. The hub does have a few issues with it, it requires a great internet speed and uses a lot of bandwidth, which causes people with a lower internet to suffer greatly on video surveillance. If the connection on the hub is not strong the video quality often lags a lot and skips frames. They state that they need a minimum 2 mbps upload speed when having seen it first hand it requires a bit more. The bandwidth on this system typically leaves customers with a significantly lower quality than their stated quality of 4K.

Finally the price of the Smart Hub, while Vivint does not offer the purchase of the cameras and the Hub separately from the system there remain a few things to point out. If someone wanted to purchase a Hub the price is $1699 that price includes two sensors and a motion detector. To include cameras the customer will also have to purchase each Outdoor Camera Pro from Vivint at a price of $400 per camera. Without financing that is an extremely hefty price to pay for someone who just wants to view their home. While Vivint is marketed as a smart home they do not allow for much versatility in their product. The only plans their customers can choose are 3.5-5 year contracts at 0-9.99% financing based on their credit. An average system at vivint that includes 3 cameras would be a monthly price of around $95/month for 5 years which prices out to nearly $6,000 with installation. For someone who simply would like to monitor their home without all the bells and whistles that is not an affordable product.

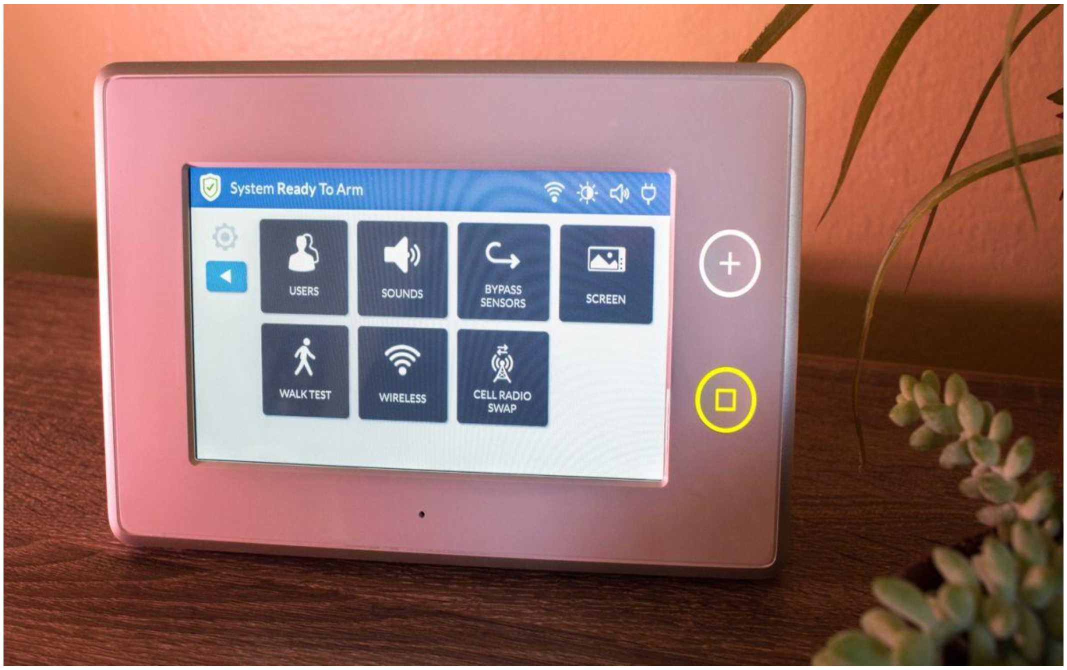
In comparison to Vivint our hub will be a lot simpler and cheaper. Our product will allow for people to connect a variety of different cameras and have a lot more flexibility. Without the pressure to buy a full $6000 security system our product will offer a great alternative to the traditional full blown security system.

3.2 Ring

Ring is a company that offers two products similar to our project design, their Stick Up Cam and their Video Doorbell Pro. The Stick Up Cam is a wireless camera with human detection capabilities that can be installed anywhere outside the premise of one's house. The Stick Up Cam records live, 1080p HD, and has colored night vision. It uses rechargeable batteries that get inserted into the bottom of the camera. Stick Up Cam starts at $99.99 per camera, and the user easily installs it themselves. The Video Doorbell Pro is a 1080p HD, night vision camera that's connected using a wire. It acts as a doorbell, so it must be placed at the door of the user's home. The Video Doorbell Pro has two-way talk allowing the user to communicate through a speaker to the person outside of the door. It offers motion detection sensors to detect when a person is at the door. Video Doorbell Pro starts at $159.99 per camera, and the user easily installs it themselves. Both of Ring's products are connected to the user's network via wifi. Neither type of camera comes with the technology for human detection or the ability to save recorded videos to the cloud. To get these features, there is a subscription that costs three dollars per month. These features are accessed and controlled through their mobile app, where the user can view alerts and live footage of the camera's recordings. The camera's recordings are accessible through the cloud for sixty days. More expensive subscriptions allow the user to store footage for longer than sixty days in the cloud. The more expensive subscription costs twenty dollars per month, and it also offers emergency assistance if somebody breaks into the home and the police need to be called to the house.

While Ring offers a great alternative to the traditional security system the system also lacks in a few areas. Ring does not have a central monitoring hub that allows the user to look at. The only way to view the cameras is on the phone with a significant delay. Ring does not have a great alert system, every camera has issues that relate to sensitivity and notifications. Many reviews on the product state that the cameras are too sensitive and often receive tons of texts that someone is outside their home when it was simply the wind blowing too hard. Our goal is to not replace the ring but simply be compatible with their system. With the Surveillance Smart hub the goal is to be able to purchase a ring and be able to view the Ring at the convenience of the hub. If the user is not staring at their phone or is occupied, when someone approaches, our Hub will change colors and simply alert the owner that someone is approaching. Ring will be a great product to work with and offer the user a central hub to view cameras on as well as an alert system.

3.3 ADT



ADT offers a wide variety of security products ranging from home security systems, home security cameras, life safety alarms, and smart home automation. We are interested in their home security cameras and their central monitoring hub because their features are related to our project. ADT offers a Google Nest Doorbell that allows users to view their front door from any location using their smartphone. It uses human detection to alert the user when someone is at their front door. It has the capability of detecting faces that it's previously encountered but will still let the user know they're at their home. Their other product that interests us is their outdoor security camera. Their outdoor cameras provide the same features as the Google Nest Doorbell, but the locations of the cameras have different pros and cons. The benefit of the outdoor camera is it has a broader range of video since it has an open area to record compared to the front door that is generally narrow in design. The benefit of the Google Nest Doorbell is it's easier to communicate with the user since they are waiting at the door for a response. Both cameras offer two-way communication, a feature controlled using the smartphone app allowing the user to communicate with the person at the door even if the user is not home. Both cameras also record the footage using 1080p HD, with night vision capabilities, allowing users to view live footage from their smartphone. ADT won't provide a price estimate until you consult with one of their sales representatives, that must first come to inspect your home.

In regards to ADT's hub, they use a similar hub to Vivint. The ADT hub lacks the ability to view and monitor the cameras. Their hub simply acts as a way to arm and disarm their system. While ADTs prices are slightly less expensive than Vivint they still do not offer the ability to monitor cameras from the hub. A basic ADT system starts at around

$65 and if third party cameras are added the system can range up to $110 a month with a 3 year contractual obligation.

4.0 Administrative Content

This section will discuss all the content that would be handled buy the lead of the company. The content in this section will include but not limited to budget and financing, marketing strategy, group management and milestone achievement, future direction.

4.1 Budget and Financing

Our budget is estimated to be around $500, this gives us a little wiggle room when ordering parts. The project will most likely cost less than this but setting a value larger than the estimated cost is important because if things need to be replaced there is room for error. Budgeting out an idea before the design is 100% complete is a difficult task because when prototyping many things can go wrong.

These parts are items that *could* be used, but are subject to change. We are currently researching the most economically efficient parts for their respective capabilities in our block diagram (Figures 3 & 2). When viewing certain hardware we selected the cheapest most effective components that will lead to a cheaper overall build. The idea behind the project is to make a security solution that could be affordable for the average consumer in the market. With that being stated we viewed certain decisions, such as what kind of display interface to use, as a modular choice depending on the specifications and need.

In order to finance this project each group member has agreed to split the cost of the components, so each person will pay ¼ of the total price. As of right now the estimated price to be paid for each group member is around $125. This was extremely affordable for everyone so self financing was the best solution. Since the price is affordable for everyone we decided to not look for a sponsor for this project. Finding a sponsor could cause an array of issues and add more constraints on the design and functionality of the project itself.

Item	**Price**	**Quantity**
Resistors, Capacitors, other small electrical components	TBA	1

LCD capacitive touch screen	$140	1
IP Camera	$39.99-$60	3
Raspberry Pi 4 8gb Model B	$75	1
Mini HDMI to HDMI converter	$5.99	1
USB Type-C	$9.99	1
3D printed case	TBA	1
DAC	$20	1
LED strip	$20	1
Total	$391-$451	

4.2 Marketing Strategy

The most important task after achieving a working prototype is marketing the product. The marketing of this product will be vital to the success. In the security industry there are hundreds of options for people to choose from so standing out is necessary. The Smart Hub Surveillance system needs to be marketed as a solution for people to have a control system for their cameras. People will also be able to customize a lot of things through the system. They will be able to use this hub to fully experiment with what their cameras can do for them. We are offering a tip of the iceberg product that allows the user to develop more on top of what we are offering them.

For tech connoisseurs this product will allow them to integrate different AI models into the system and allow them to fully expand upon what we are giving them. For the average consumer we are offering them a piece of mind. People get tired of staring at their cameras all day to see what's going on around them, with the smart surveillance hub we are doing all the hard work for them.

The first marketing related thing we would need to do is approach camera companies and try to do a partnership with them where they allow us to offer a package to sell the hub with their cameras. If someone went to the store and saw if they could purchase the smart surveillance hub and two discounted Ring cameras came with it, it would be a no brainer for a lot of people. Aligning ourselves with well established surveillance companies is the first goal.

Furthermore, getting our product into a variety of stores would be crucial. Selling the Smart Surveillance Hub at Target and Best Buy would be ideal because those are where tech connoisseurs and our target customers will be. Implementing these strategies will result and a successful product roll out and help with the long term success of the product.

4.3 Milestones

The tasks listed in the table state the goal and the date we want to achieve each goal. Each group member is responsible for coming each week with their required work done and ready to be discussed among the group members. Most of the milestones have been hit and the

4.3.1 Senior Design 1

Senior design tasks include discussing various ideas. We started at creating a mesh radio network, moved to a drone, then to a threat deterrent camera, and finally landed at the smart surveillance hub. Each idea required a significant amount of research and discussion to determine if it was a viable/possible option for all of us to complete given our skills. Once we finally landed on the Smart Surveillance Hub we all had to do research regarding what our idea involved. We then separated the project into different tasks for each group member to complete each week. Every week we showed up and discussed research that we had discovered and problems that arose. This helped us problem solve together and interface the hardware ideas with the software possibilities. Completing each task in a timely manner was vital to the project design. Moving forward without having a complete design, into Senior Design II, could cause many different problems and result in a not functioning prototype.

Number	Task	Start	End
1	Think tank - Ideas	01-17	01-24
2	Narrow down options, design parameters and select a project.	01-24	01-13
3	Initial Document - Device and Conquer	01-31	02-03
4	DC1	02-04	
5	Review and discuss our project with Lei Wei	02-07	

6	Investigate IP protocol networking	02-07	02-10
7	Obtain an off-shelf generic IP camera	02-10	02-14
8	60 page Senior Design draft	02-14	03-25
9	Explore options for software development	02-17	02-24
10	Determine and choose final parts for PCB	02-24	02-30
11	Assignment of standards	02-28	03-11
12	Test and configure software compatibility between the Pi4 and the IP camera.	03-01	03-10
13	Design the PCB and have it made	03-10	03-21
14	Acquire additional finalized parts	03-10	03-27
15	100 page Senior Design report	03-28	04-08

Table 3: Milestones

4.3.2 Senior Design II

The Senior Design II tasks discuss a variety of prototyping steps and tests that need to be taken to ensure the completion of the project. Each task needs to be completed in a timely manner in order to ensure that the prototype is tested and working as designed. Many tests are going to need to be run before the final product is built. The most difficult task to complete before the assembly can be completed is the software and hardware configurations. Every step past that requires heavily on the software and hardware working together properly. Each step after that is just adding a little bit more complexity to the initial task. To complete this in a timely manner each group member needs to work diligently to handle their respective tasks. The Electrical Engineers need to have the PCB designed and the hardware tested so that when it is passed to the Computer Engineers it is working properly and easily coded.

Number		**Start**	**END**
1	Test Software to hardware configuration	04-09	04-30
2	Develop software for the Smart Notifier	05-01	05-10
3	Develop software for the Hub application	05-11	05-30
4	Design a 3D printed case of AutoCad	06-01	06-10

5	Print 3D printed case	06-11	06-15
6	Stress test the finalized product before final assembly	06-16	06-20
7	Assemble the Smart Hub	06-20	06-22

5.0 Project Summary and Conclusions

Resources

[1] Federal Bureau of Investigation. (2019). *2019 Crime in the United States: Burglary.* https://ucr.fbi.gov/crime-in-the-u.s/2019/crime-in-the-u.s.-2019/topic-pages/burglary

[2] ADT. (2022). *Is a Wireless Security Camera Worth the Cost?*. https://www.adt.com/resources/wireless-security-cameras-cost

[3] Raspberry Pi 4 Specification, Schematic, Diagrams https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/

[4]https://lucasteske.medium.com/reverse-engineering-cheap-chinese-vrcam-protocol-515c37a9c954

Displays considered

[5]https://www.amazon.com/SunFounder-Raspberry-Touchscreen-1024%C3%97600-Capacitive/dp/B07Y889J3X/ref=sr_1_1_sspa?crid=1GCWRIR1GYHGO&keywords=touch+screen+lcd&qid=1643949017&sprefix=touchscreen+lcd%2Caps%2C170&sr=8-1-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUEzTzNGNUpSRDVPQUwmZW5jcnlwdGVkSWQ9QTAzMjQ4MzczUFhMTzdLSVQ0OTlKJmVuY3J5cHRlZEFkSWQ9QTA1Nzc0MDcyRDhCMlZZOUJVOFREJndpZGdldE5hbWU9c3BfYXRmJmFjdGlvbj1jbGlja1JlZGlyZWN0JmRvTm90TG9nQ2xpY2s9dHJ1ZQ==

[6]https://www.amazon.com/dp/B01J51CXU4?aaxitk=b9e084d794b47684d8c57912dc75bb6c&pd_rd_w=rdGLv&pf_rd_p=81374be5-e348-4c15-938f-a311a861c514&pf_rd_r=SCWKQN14V60DNC4J6PB4&pd_rd_r=cc3846ad-842f-4569-9b98-a7740adcc87b&pd_rd_wg=oOgte&ref=dacx_dp_4281262610001_5940343004497700111

Camera considered

[7]https://www.amazon.com/Amcrest-5-Megapixel-NightVision-Weatherproof-IP5M-T1179EW-28MM/dp/B083G9KT4C/ref=sr_1_1_sspa?crid=36SJVIR898Y67&keywords=ip+cameras&qid=1643951080&sprefix=ip+cameras%2Caps%2C135&sr=8-1-spons&psc=1&smid=AM1AKWRN957PC&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUFDTDNEN0g0OUtCSTUmZW5jcnlwdGVkSWQ9QTA3MzgyNTVUMU1VRDkwVUM5RlUmZW5jcnlwdGVkQWRJZD1BMDAxMzIxOTJYWU1CWVIzTjlGRTAmd2lkZ2V0TmFtZT1zcF9hdGYmYWN0aW9uPWNsaWNrUmVkaXJlY3QmZG9Ob3RMb2dDbGljaz10cnVl

[8]https://www.amazon.com/Wireless-Security-Wansview-Detection-Compatible/dp/B07QKXM2D3/ref=sr_1_5?crid=2XWKJ037NQ2P8&keywords=ip+camera&qid=1643951179&s=electronics&sprefix=ip+camera%2Celectronics%2C131&sr=1-5
[9] Vivint
https://www.vivint.com/ppc/brand?cq_src=google_ads&cq_cmp=146236557&cq_con=7629216237&cq_term=vivint&cq_med=&cq_plac=&cq_net=g&cq_plt=gp&utm_medium=cpc&utm_term=vivint&utm_source=google&c_ps=s.google_k.vivint_m.e_n.g_po._d.c&exid=117156&geo=9053023&gclid=Cj0KCQiA64GRBhCZARIsAHOLriIg_qJ0Kg_h_ZH3ORTHyE4MRUnbkoV8cjiA0kLyc_uWDAdT0i54SyUaAtDzEALw_wcB
[10] Ring
https://ring.com/collections/all-products
[11] ADT
https://www.adt.com/security-cameras
Software Testing frameworks
[Jest] https://jestjs.io/