Pegasus Protection Services

Senior Design 2

Group 36

Final Document

University of Central Florida

College of Engineering and Computer Science

Dr. Lei Wei

Dr. Samuel Richie

Isaiah Williams	Computer Engineering
Christian Silva	Computer Engineering
Dylan Sauerbrun	Computer Engineering
Aundre' Fredericks	Computer Engineering

TABLE OF CONTENTS

8. TEAM RELATED

9. PROJECT SUMMARY AND CONCLUSIONS

APPENDICES

Total: 139 pages

LIST OF FIGURES

LIST OF TABLES

1 Executive Summary

Bluetooth 4.1 and Bluetooth 5.0 are used in everyday technology. Over the years, Bluetooth has evolved to become smarter as the applications of the technology become more diverse. There are new incentives when it comes to utilizing the increased range of Bluetooth. As Bluetooth grows, so could its applications in the implementation of security. Businesses with secure sites have added many layers to internal security through training employees on phishing, requiring keycard access to enter certain buildings, and utilizing security cameras. There must be an even smarter way to track employee/non-employee activity within a secure site. Our goal is to take the capabilities of Bluetooth and utilize them to engineer a Bluetooth tracking system along with the use of computer vision.

When completed, the Bluetooth tag tracking system will have the functionality to add another layer of security to what already exists in many large corporations. What will this mean? Along with allowing users access to buildings, the Bluetooth tag will communicate with in-range antennas to pinpoint the user's real-time location within the building. In addition, there will be a companion app developed to visualize user info and their location history.

In use of the application, there will be user privileges applied when a user is granted a Bluetooth tag. On-site security will need access to ALL facilities within the building, Executives may need access to ALL sites within the building, but everyone else's access will be based on their respective department and special access depending on their roles. These items will be managed within a secure database, intended only for database managers to access. One exception to be considered is providing guests access to the secure site. These individuals will be allowed to access facilities with temporary access. Another potential feature is executives/managers having access to activate/deactivate department guest Bluetooth tags through the companion app without having to go through security or IT to get the access granted.

Generally, Bluetooth modules and Computer Vision components are inexpensive. Our goal is to implement these Bluetooth tags in conjunction with Computer vision. An individual walking into a room with a Bluetooth tag will first be identified with the cameras in the room. Each camera will search through a database of individuals working for the company. If a person working for the company does not have a picture on file yet (they have a Bluetooth tag, but no image tied to the tag), the system will send a flag to the security personnel. Another scenario could be an individual that could just be visiting the site for business transactions, maintenance of equipment, etc.… These cases will search the database and automatically snap an image of this individual. The signal from the Bluetooth tags, communicating with the security cameras and antennas will be transmitted to the companion application and displayed on security's portal. The real time location will be transmitted by the circuit holding the antenna technology. The location transmission will only activate once the user has scanned their tag to enter a facility. Our goal is to have the ability to log user's location with accuracy to the room they are within and their timestamps.

The Bluetooth tag is intended to protect users as well as company information from being stolen. If company information is outsourced, authorities would have access to information that could save them time and resources. Data that belongs to one facility could be tracked easier because they would have access to anyone within facilities and the times related to the data accessed. The tags would eliminate false alibi's and hold individuals accountable when proprietary data is mishandled. The end goal of the Bluetooth and computer vision security system is to add more integrity to our modern security systems.

2 Project Description

Section two covers a comprehensive view of the project and the functionality. In this section team motivation and goals, project objectives, requirement specification and use case diagrams will be discussed in depth to show why our team selected this project, what we wish to accomplish and to give an outline of what the smart security system is expected to do. Reading this basic overview of the entire project will help give a better understanding of the technical content.

2.1 Project Motivation and Goals

The primary motivation for this project was to encourage every team member to step out of their comfort zones and work on a project that tested us in fields we have not had much exposure to. The smart security system was the perfect project to do this. All members of the team have had exposure to both hardware and software throughout the workload of the undergrad courses for computer engineering majors. All members of the team made it very clear that through the duration of their undergrad studies, that they felt more comfortable with software than hardware application. Due to this reason we decided right away that we wanted to explore the software area of computer vision, and we figured that creating a security system would be the best implementation of this technology. Using computer vision, we plan on creating a facial recognition system. There are also no software courses for undergrad students that heavily teach facial recognition. Of course, the team made it very clear that we all wanted to use this project to introduce us to the challenging world of hardware, which is why we wanted to incorporate some sort of indoor tracking system. There are no explicit hardware design courses for undergrad students that help you learn how to implement indoor tracking or the incorporation of the wireless technology of Bluetooth. This project fulfills our desires to learn these new technologies, create complex designs, and essentially helps us step outside our comfort zones. This project serves as a milestone for us, as it connects our academic careers with our future professional careers as engineers as we strive to learn effective teamwork and expand our knowledge of hardware and software systems.

The primary motivation of this project is to protect exclusive information of companies with the use of computer vision and Bluetooth indoor location tracking. Information is protected by eliminating false alibi's and holding individuals accountable when proprietary data is mishandled. Our team knew right away that we wanted to come together to help tackle problems with current day security systems. The team knew that no matter what we wanted to incorporate computer vision in our project. The deciding factor that led to us selecting this project was the addition of Bluetooth indoor tracking. Every member of the team showed interest in wanting to learn how indoor location is tracked and decided to implement this to the project, overall creating an all-in-one smart security system that has facial recognition cameras and the ability to track an individual throughout the building.

The primary goal for this project is to build a functioning prototype of a smart security system that utilizes facial recognition software to detect individuals within building location and to design an accompanying Bluetooth tag to assist with indoor location tracking of users through high secure areas of the building. We choose this project because of its complexity and the challenge it brings by taking the team out of the comfort zone. This project must be feasible, fit within an appropriate budget, and be able to be completed within our time constraint window. Using these guidelines, the following prerequisites were created. Firstly, the team must create a database to store facial images and videos. Storing these media files will provide us the availability to be able to retrieve information of facial records from the database to match to the corresponding target the camera detects. Secondly, the development of a web application is crucial, building the web application will allow users to view, add, select, and delete target users. Next, the setup of the camera is crucial, which will allow analysis of objects that come within the field of vision. Following this, Bluetooth beacons must be placed within a secure area location. The beacons are responsible for picking up the tag that users will have when entering these areas, and it is the foundation of our indoor location tracker system. To summarize, using these guidelines, the team is able to split the dedicated goals to come together and create the smart security system.

Ultimately, the goals of this project are challenging and complex but reasonable to complete. We have noted in the past that previous projects in senior design have had multiple implementations of computer vision/facial recognition software, there has also been a project in the past based on Bluetooth indoor tracking. The uniqueness of our projects comes from everything being put together to create a security system - by combining aspects of computer vision and Bluetooth tracking to make our system align with our expectations.

2.2 Objectives

The goal of this project is to create a camera system that analyzes live feed for when people walk within the field of view. The camera is able to pick up the target and put a face to the name to successfully identify people and to see whether or not that individual has the right to be within that secure area of the building. Another goal is to successfully create a Bluetooth tracking system, in which users will be given a Bluetooth tag that will be picked up by Bluetooth beacons that will be setup in high secure areas of buildings, such as servers farms or any area with proprietary information. The beacons will be responsible for picking up the tag location, relative to the placement of the beacons, so we must make sure that the beacons are placed properly to cover the most amount of area in the room. Combining both these goals together, should leave us with a system that uses a camera to detect targets within the frame, if the target is not allowed to be in the secured area, then a notification will be sent out to the building security and executives of the trespassing going on. If the target is allowed to be in said area, then once the user comes within range of the beacon, the tag and beacon will start transmitting location within the secure area, relative to the beacons.

2.3 Requirements/Specifications

The project will consist of four major components:

1. BLE Beacon: These stationary beacons will be evenly placed in room to receive positional data from tag and send this information to the security software (PCB requirement made here) can be made relatively cheap

2. BLE "asset" tags: These tags will be on the moving employee/user and will be constantly transmitting their positional data out to beacons

3. Software Application: This software will give administrators access to database of facial scans as well as provide an interface for live tracking of users in secure area

- Software will also contain employee's location history with timestamps and path they took while in secure area
- Will also receive any warnings triggered by the security system detecting intruders or unidentified users
- Will allow for employee tags to be set/changed to BLE tags

4. Pan-Tilt-Zoom Cameras (made with servos): Main functionality of these cameras is to fulfill the identification requirement, which will be done through facial recognition. The goal of the cameras is to be Pan-Tilt-Zoom capable.

- The pan-tilt function will come in handy when establishing sight of user to identify the face while user is moving into and around secure area
- Ideally the camera will pan and tilt with the user's movements to perform faster and more accurate recognition than just a stationary camera mount
- This movement tracking can be implemented with location data from BLE tags or just through recognition algorithm

2.4 Component Specifications

BLE Beacon:

- Will need to make multiple 2-4 beacons for tracking data
- Will need to be able to send/receive position data from BLE tags

- Device just needs capability to receive Bluetooth data and then transmit it to Program that will process this data
- Beacons will be Microcontrollers, with the model being an Espressif ESP32
- Will be powered with PCB power supply
- Will have the ability to be wall mounted
- Capable of UART communications with other beacons wirelessly
- Will be an edge device on systems cloud service and will provide constant live updates to cloud databases.

BLE "asset" tags:

- Tags will also be represented using ESP32s that are designated specifically as location tags
- Will be able to transmit accurate positional data to receiver beacons
- Ideally should be small enough to mount on employee ID card or placed in pocket
- Beacon will be broadcasting data to at least three wall mounted beacons.

Software Application:

- Easy to navigate interface
- ability to upload area/room layout to show accuracy
- Purpose of the application is to provide an interface in which the live positional data of the user is displayed as well as an accurate trace of their movements throughout the room.
- Database should be encrypted
- BE ABLE TO ENABLE/DISABLE certain user access remotely
- Application will be able to show data for any user stored in this database
- will display the time that transmission was initiated as well as show an interface with the path the user took.

Pan-Tilt-Zoom Cameras (made with servos):

- Cameras will be mounted on servo motors to allow for a wider range of motion
- Cameras will either have a physical zoom or digital zoom component to make facial recognition more efficient and minimize detection times
- Camera will make use of beacon data to determine
- Camera will be mounted with one or two servos depending on implementation
- direction control done through a simple microcontroller

2.5 House of Quality Analysis

Below in **Figure 1** is an illustration of our House of Quality diagram; it shows the aspects of our project that our group found important. Note that the targets for each engineering specification is a rough estimate and will become clearer as time goes by. This diagram shows our primary focuses. We want a system that minimizes cost while maintaining good reliability, which is why we are willing to sacrifice the size of beacons and the power consumption.

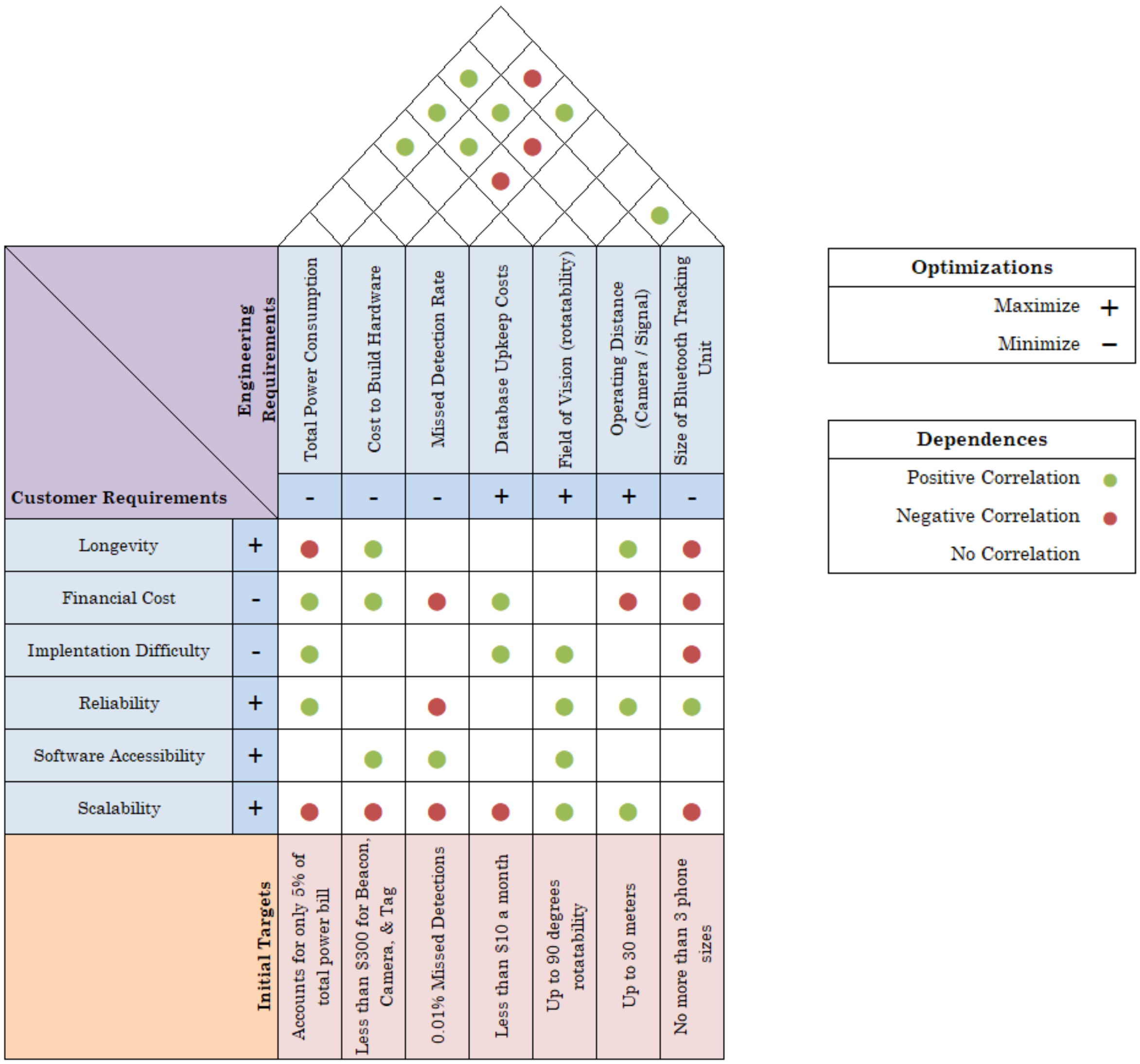


Figure 1: House of Quality

2.6 Initial Design Architectures and Related Diagrams

2.6.1 General Project overview:

Figure 2, seen below, displays the basic idea of our project. We basically are creating a security system that keeps track of indoor location using BLE. Doing this will allow us to acquire real time data of where the tags are when they enter into a highly secure area. As you can see below, depending on where the user is trying to go within the building will determine if the tags start being tracked by the beacons we have set up. If the tag is picked up in the secure area, then the BLE tracking system will kick in and start storing the information into the database, deploying the information to the admin center we plan to create.

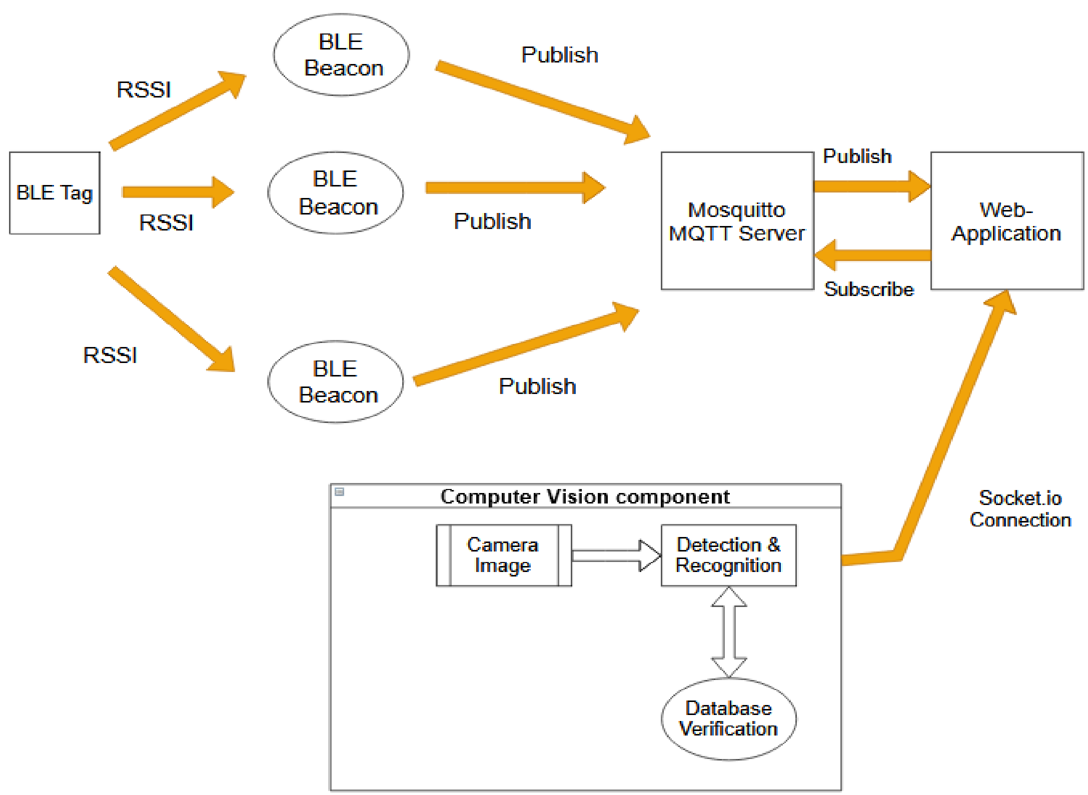


Figure 2: Project Overview Block Diagram

2.6.1 Hardware overview

We plan on splitting up our projects in two parts, hardware and software. For the hardware side of everything, we have a basic idea of what we are working with, but of course more research needs to be done. We are using BLE tech due to them being less expensive and previous research done on this topic. The beacons/antennas are the most important part of the hardware. The beacons will have to produce a signal strong enough to pick up the Bluetooth tag in any location within the secure room. Below you will see the basic general workflow of the hardware side of this project. Isaiah and Aundre will be the ones primarily responsible for the hardware aspect, Christian and Dylan will be assisting as well with the BLE tracking enabler and database system to store the information. Everything in figure 3 still needs further research.

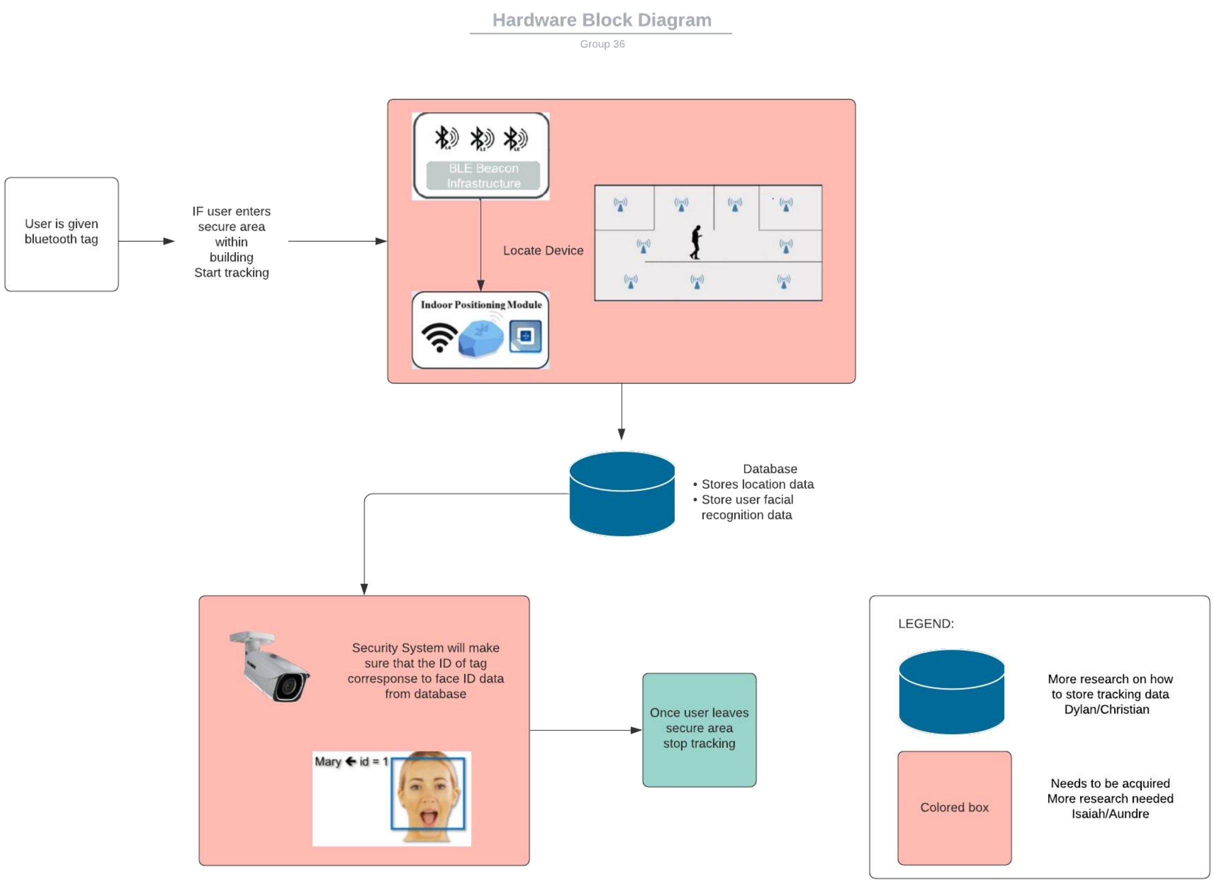


Figure 3: Hardware Overview Block Diagram

2.6.2 Software Overview

The second aspect to this project is the software side. We want to make this data accessible for users who have admin privilege and oversee the building, for example - security, building manager and executives. Christian and Dylan will be primarily responsible for this aspect of the project. We want to create an admin center where users have to login to get access to the trackers data. From here the admin has access to view tracking data and to assign/remove tags from corresponding members of the organization. Figure 4 displays the workflow for this particular application.

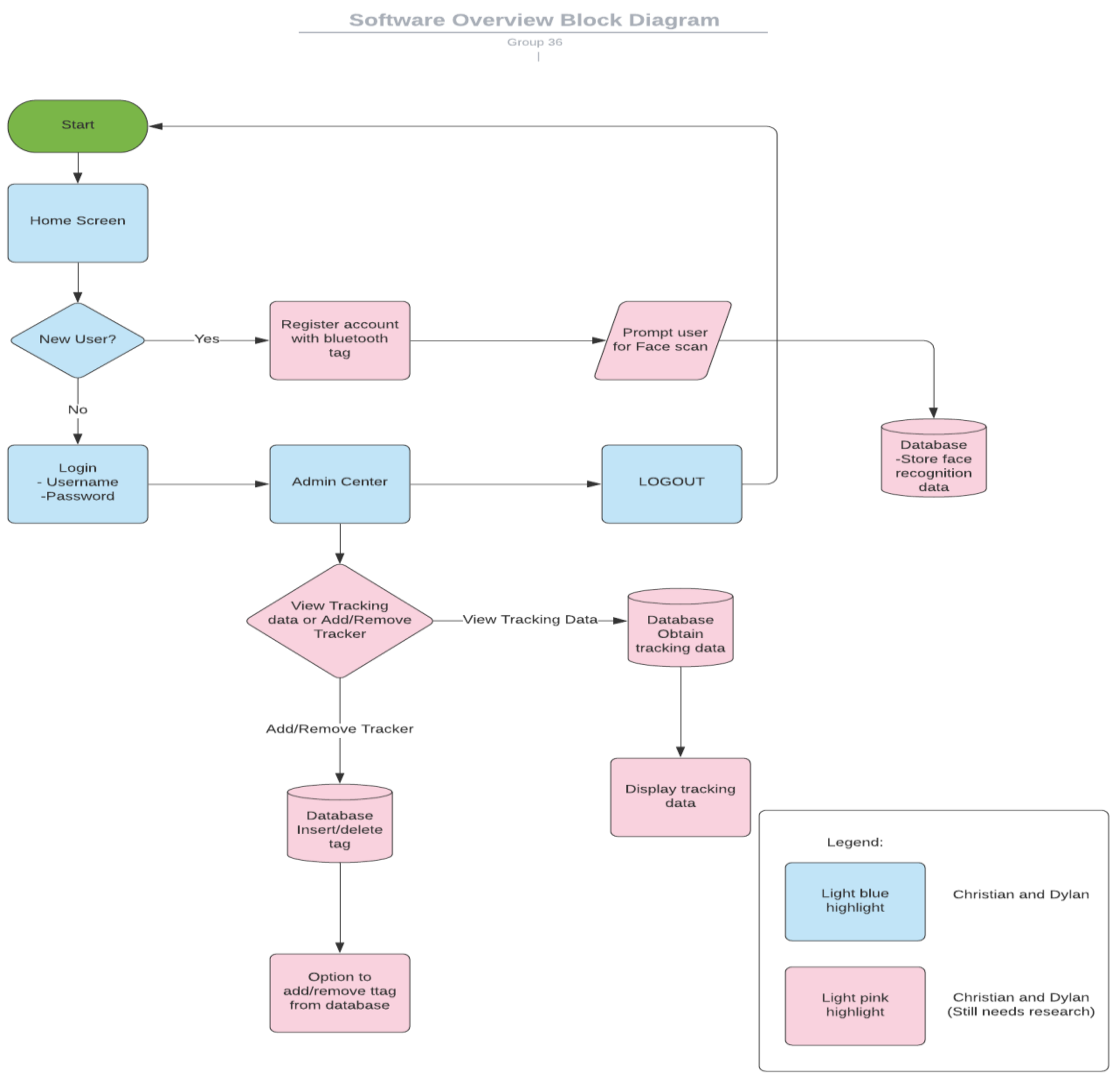


Figure 4: Software Overview Block Diagram

3.1 Existing Similar Projects and Products

For research we went searching for projects that have been done previously by ECE senior design students and of course from other schools within the national border. Indoor tracking is a relatively popular idea that has been around for a while, although when initially searching we struggled to find good tutorial videos, our team did find plenty of resources that guide us down the right path for our project. As mentioned before even though it is not straight forward, we believe that we are capable of creating a successful indoor tracking system. From searching previous projects from the University of Central

Florida ECE senior design site, we see that indoor tracking has only been done once before in 2014. However, we did find plenty of other projects that align with our other system, which is the facial recognition camera, but currently there is no project that incorporates both together due to its complexity. The following examples are similar research projects we came across that we found useful while researching.

3.1.1 Indoor Positioning with Bluetooth Low Energy (BLE)

This project was made by instructables user: cincoutprabu. Its purpose was to be a proof of concept for a Bluetooth-centered indoor positioning solution. This user made use of intel Edison boards, Arduino boards, TI sensor tags and their mobile device to create a system that is capable of determining proximity from a node using only Bluetooth signals. For the most part this post is a demonstrative one in which this user explains the steps they took to connect the tags through a software to establish connections. The intel boards are used as “base stations' ' (beacons) that will remain stationary and provide a reference point from which the Bluetooth proximity can be measured. While this project lacks the use of direction tracking it did provide a baseline example for our group to build on. From this project we can see that Bluetooth beacons placed around a room can provide us with accurate proximity data and with alterations to the system can provide both distance and angular position around the beacon.

3.1.2 Autonomous NERF Turret with Facial Recognition (ANT-FR)

ANT-FR was created by Steffen Camarato, Nicolas Jaramillo, and Michael Young. The autonomous NERF Turret with Facial Recognition or ANT-FR is a camera-based weapon system that uses facial recognition software that analyzes live camera feed to detect targets that come within the camera's field of view. If the target is detected within the frame, the turret will shoot at it until the target is no longer visible by the camera. Although this project's overall objectives and goals are different from the system we are trying to build here, this turret system relies heavily on facial recognition as a key component, similar to ours.

The ANT-FR needed to use facial detection to see if a target entered the field of view of the turret, if so then the turret must fire at the target until it is no longer visible. Their facial recognition acquired information from a database of faces it collected as “potential targets” and if the camera detects the target, it will follow the target and fire at it. Using OpenFace, they were able to implement this software system by adding annotations around key features of a face with 68 reference points, and by training the algorithm by random model initialization. By using their mobile application, they developed, they were able to select/deselect targets for the system, which is also a feature we wish to add to our mobile application [1].

Although our team plans on using OpenFace as the part of our resources for facial recognition and plan on going about training the way they did to have it optimized for live video feed, we do plan on using the DLIB library to train our system, which encompasses both support vector machines and Histogram of Oriented Gradient to achieve better accuracy. We also plan on using a MERN stack to develop our web/mobile application as it is more used in modern day full stack development. Overall, this project has successfully completed a similar facial recognition system we wish to incorporate into our project. They did run into some trouble with the training and accuracy of the system, but we hope to change that around by using different techniques to train our model.

3.1.3 Indoor Navigation System with Beacons

The Indoor Navigation system with Beacons was created by Andre Compagno, Josh Facchinello, Jonathathan Mejias and Pedro Perez. In this project their group had the objective of creating an indoor navigation solution based on BLE beacons and tags. Their project's Bluetooth component is very similar to the way that our group plans to implement it, that being that BLE will be used to provide real time position of the user as they move through the environment. Their project involves using google glass to provide a way for the user to receive real time directions as they move through this space.

In their project design specifications this group talks about the need to create their own BLE beacons including BLE antennas in order to have a relatively cheap Bluetooth solution. The beacon they use has two modes: Advertising Mode and Configuration Mode and these modes allow for location data to be advertised and also to be altered respectively. These communication modes are supported with the UART communication protocol. Their beacons make use of the iBeacon protocol as well.

The software component of their project is done through either an android or apple mobile app, whereas ours will be done through a web-only application. Their software allows them to debug various parts of the setup, such as altering the database or parsing the values of the configuration values being sent by the beacon. To represent building layout this group made use of a unique JSON format that represented different attributes of the building in which they operated in. Using an array, they show obstacles in the way as "-1" and the rest the number it according to the 2d array size.

The use of BLE means that a developer must also consider the fact that there can be many other BLE devices around, so filtering is necessary to ensure that the correct device is being connected to the beacon. And to derive the distance of the Bluetooth this group utilizes RSSI and further steps to ensure that the distance is accurate in relation to the location of the beacons. And then to find location this group makes use of trilateration, as they only had access to Bluetooth 4.2. Our group will have access to Bluetooth 5.2

which introduces Angle of Arrival which provides a much more accurate framework for determining position of a user. While the actual implementation may differ due to new Bluetooth technology, the idea in goal is the same for both this project and our own [2].

3.1.4 Existing Products

In production, many indoor positioning systems have been implemented. Of those, a YouTube video was one of the existing products that helped our group believe that we could implement our prototype. In this video, “estimote” developed a prototype that could accurately track an individual indoors as long as they were holding on to the tag. For the demonstration, the beacons were placed throughout a lobby hosting many individuals also watching demonstrations. As the demonstrator moved around the room, tracking was visible in the application developed alongside the beacons. Though indoor security is at the root of both the existing product and our product, there are some key differences that are affected by time constraints and economic constraints. Our budget for this system is not nearly as steep, therefore, the implementation will not be as robust as the “estimote” prototype.

To give an overview about estimote, they are one of the biggest implementers or indoor tracking systems. Using various technologies such as Ultra-wide band and Bluetooth low energy, they have several implementations on location and proximity solution workspaces to increase safety and productivity. They provide an extensive IoT service for beacon analytics to your application, so you are able to see everything functioning in real time. They also have extensive libraries that go into detail on how to enable beacons, disable them, add in unique tags for employees and go over how to authorize location data for usage across the workspace, delivering an all-in-one solution for an indoor Bluetooth low energy security system.

A really well known and popular accurate indoor positioning system is Pozyx. Pozyx uses Ultra-wide band technology because of its increased accuracy. Using Arduino based systems they deliver this indoor solution. UWB delivers low power, its signals can penetrate through objects and thin walls, and it is easy to implement in any building. Similarly, to estimote they have accompanying software that shares real time analytics about beacons and its corresponding tags to see where individuals have been, add or delete any existing user tags. This company plans on being the biggest indoor positioning company in the world and with their RTLS tracking based on UWB technology, they are on that track.

An undergraduate team at the University of Colorado implemented a tracking system much different than our current standing project, but similar to the first iteration of the team's idea. Our initial product involved utilizing RFID to track individuals in an indoor setting. The undergraduate Senior Design teams project wanted to track real-time student access within university facilities. The main goal for this group was to add a level of security to the safety of the students accessing University facilities, but also protect the University facilities from threats of crime. The reason this project is no longer similar to the current idea is because RFID was removed entirely. Implementing Bluetooth Low Energy technology is the better route for this project because we can connect wireless devices without having to put them into physical contact with one another and we can expand the field of vision amongst the wireless devices communicating with each other.

3.2 Relevant Technologies

This section describes the various technologies that were investigated for the overall design process. To fully understand the scope of our project, this section will cover the software and hardware background research we explored to give the users a better understanding of the smart security system and its desired functions. While some technology we go over will not be utilized it is important to understand the concept behind them and why they were either considered or used as a steppingstone for our project.

3.2.1 Software Technologies

As stated before, the goal of this project is to create a smart security system that is capable of indoor proxy tracking and develop a pan-tilt camera system that uses face recognition to identify users who enter high security areas. In this subsection we go over the relevant technologies that will help us create this facial recognition system as well as an accompanying web and mobile application that users can use.

3.2.1.1 Computer Vision

Some of the reasons why we elected to bring cameras into our system stems from the idea that, in reality, the trackers work efficiently when used in instances where 1) everyone is responsible and wears their respective cards, and 2) no other person besides the employees are going to be in the area. Unfortunately, the trackers alone beg the question: What if an intruder decides to break into the work area? This situation requires some means of determining when it occurs, and since the intruder is likely not going to have a tracker (it is probable they do not work there), it would be pretty easy to slip by without the help of a camera.

Now that we understand the reasoning behind using a secondary device such as cameras to enhance the reliability of our system, we should discuss how we plan to identify faces on the cameras. We could simply hire people to exclusively manage the cameras during work hours, but in this day and age, that approach does not adequately make use of the useful tools available to us. Enter the concept of *face detection*, which employs computers to oversee the many images that are provided by the cameras and pick out faces from them. The idea would be to use one computer as the 'eye' for all cameras in the work area, and when inconsistencies arise, have it relay the message to a real person. Really, the face detection concept is considered a subdivision of the object detection technique, where the computer seeks to identify the object specified throughout the field of images. Computer algorithms are often used to implement object detection, and the terms for which the technology will work on must be maintained in order to keep the process dependable.

Once we have the faces detected in our system, we need methods for mapping the face to one that is either an employee or not. There are two routes we can go in terms of implementing this in our system. The first would be opting to use face detection in conjunction with ID card (object) detection that will both indicate faces on the camera and seek to recognize an identification card that would be on a person’s torso and use that to determine whose face is being perceived by the camera. The major downside to this approach is that employees working must be wearing their ID card at all times so the camera can properly identify them. The other path we can use to achieve the same goal would be to tag on a facial recognition algorithm that will use the images the camera is seeing and match it to an employee based on a collection of employees’ faces on a database. By using this approach, there would be no real need for the ID card to be displayed at all times. With that in mind, a fruitful amount of time was spent researching the various methods for sufficiently implementing face detection, facial recognition, and object detection (for the card). After comparing the techniques for each feature, we have elected the ones that seem plausible to pursue within the given time frame.

Facial Detection

Facial Recognition

As briefly mentioned above, face detection refers to the concept of recognizing a human face within the pixels of a digital image. The task is strictly observing when faces show up in the image; then, to take the faces identified and map it to an actual person's face, facial recognition is employed. We go into that later. Moving right along, the two prominent strategies for detecting faces are *image-based* and *feature-based*. What do these terms mean? Image-based approaches rely on training a machine on a dataset of face images and then putting the now newly taught machine to the test with images that it has never seen. On the other hand, feature-based approaches use a machine's understanding of facial features and seek to observe the same features from the images passed to it. Both approaches have its benefits and drawbacks, which we get into after covering each individually. Approaches for face detection as follows, as per [31]:

<u>Image-based Strategies</u>

- *Neural Networks*: A widely talked about method in the machine learning field, neural networks are machines that are essentially trying to imitate the human brain and build an interconnected system of neurons via nodes. Generally speaking, neural networks have 3 components: the input layer, the hidden layer, and the output layer. The input layer aims to gather input trends and instantiates weights, the hidden layer attempts to extract and refine the input and weights so that the overall error of the model is negligible, and the output layer attempts to provide likelihoods for each particular label or outcome, producing the answer as the highest of them all.

- *Statistical models:* A great number of techniques in machine learning directly follow and work alongside statistical procedures in order to operate. Really, before machine learning there was only statistics. The two statistical methods we'll discuss here are principal component analysis (PCA) and support vector machines (SVMs). Firstly, PCA has the primary objective of decreasing the overall dimensionality of the observed dataset all the while preserving the variance in the dataset. In this case, PCA would be used on the training set to create eigenvalues/eigenvectors, and new faces are evaluated based on the eigenspace, which houses the collection of eigenvectors that were produced. A face image may be detected after it has been projected onto an eigenspace and measures of the distance between the subspace and image region is computed. As for SVMs, they work behind the basis of structural risk minimization, specifically aiming to balance the intricacy of a model and their fit to the training data, which averts overfitting of the test set. One such implementation of this method comes from Hyungkeun Jee, et al. which uses a SVM to confirm and select an instance of an eye pair. From there, it normalizes and pulls out a face candidate boundary based on the centers of the eye pair, and the face is validated using another SVM.

Feature-based Strategies

- *Movement:* From video records, data on the motion allows for the locating of moving objects. By thresholding the frame differences, moving face and body parts may be identified, as well as facial features.

- *Edge:* Studying the line drawings of faces from photographs brought forth the idea to trace the outline of a human head. From there, a set of procedures were developed to detect faces from edges alone. The first step involves adjusting the image via histogram equalization and application of a filter for noise removal. After, the image goes through another transformation which employs the use of the Sobel operator. An edge detection algorithm is used to generate sub windows from the modified image that is represented in edges.

- *Greyscale:* An image must first be adjusted via grayscale to provide support of the model's capabilities. Since lips and eyebrows often come out as darker than the other features in an image, algorithms are put to work at identifying these darker areas. With the darker areas being accentuated by a contrast-stretch, a model would simply need to conduct low-level gray-scale thresholding to bring out the locations of a face.

- *AdaBoost:* The idea of AdaBoost being used on a machine learning model often comes with the goal of swiftly training a classifier. To do this, the algorithm selects visual features from basic classifiers and aggregates them linearly. Through a set of weak experts, examples will pass through each individually, until one expert deems that example as invalid. The idea behind combining a set of weak experts comes with the goal of being able to quickly validate a set of examples and works upon the premise that faces are rare to show up in the observed image. The biggest drawback with AdaBoost is that it is quite sensitive to outliers and noisy data.

- *Feature Search:* Off the back of a newly innovated method of object detection that reduces overall calculation time, the Viola-Jones object detection framework appeared with claims of significantly improving on the speed at which faces are detected (for its time) while keeping near perfect accuracy. Utilizing the Viola-Jones approach to face detection, Haar-like features, or a collection of dynamically scaled square shape patterns, are qualified rather quickly due to the use of the Integral Image as the form of image representation. With the help of AdaBoost algorithm, this face detection structure is to be used on grayscale images in a scanning manner. Finally, the sub-windows that are not thrown out by the opening classifier go through Because of its ability to quickly evaluate faces, many would say that the Viola-Jones method is most practical for real-time face detection. However, weaknesses have been identified as having an extended amount of training time, and not reliably working on darker-skinned faces.

- *Local Binary Pattern (LBP):* Like the Viola-Jones approach, LBP works at a relatively fast rate, but also provides rotation invariance, allowing for it to be applied in the broader field of image retrieval. For each pixel in the image, a texture value becomes associated with it. It uses background subtraction to perceive a moving object in the frame. Those who are interested in sufficiently describing the image texture may seek to use this approach, as it is effective at achieving that goal. Comparing it with Viola-Jones and others that use Haar-like features, it is found to be simpler in terms of the computation intensity. Some things to note, however: LBP can only be applied on gray/binary images, and crumbles in scenarios where the shifts in illumination are not monotonic.

- *RGB color model:* The RGB Model is the most fundamental of its kind and all other color models stem from it. Colors in the pixels of the image are broken down into three color components, Red (R), Green (G), and Blue (B). The model utilizes a histogram to determine which pixels are of skin color, and changes in brightness can be accounted for by correcting with the luminance. However, the model is very much susceptible to changes in light. Another detriment comes at the nature of the color model, which is that there are some scenarios where it cannot accurately separate the color from the brightness of said color. The model employs the AdaBoost method to aid in segmenting and thresholding skin color.

Ultimately, with today's current advancement in machine learning and computer vision, we have a variety of different strategies and frameworks that we can use to implement facial detection within our system. Some popular machine learning and computer vision frameworks are TensorFlow, OpenCV, and OpenFace. It comes down to choosing what will best fit our group and our needs, and we have gone more in depth about these different options in the Strategic Components and Part Selection section.

3.2.1.2 Web Application Stack

Web Development stacks are bundles of software components and languages used to create web applications and mobile applications. When developing these applications, it is important to consider what technologies will be most beneficial for our project. The most crucial part is how the storage of facial recognition data will be stored and whether we want a relational or non-relational database. We have two options for which web development stack to use, LAMP stack or MERN stack.

Lamp stacks are the most beginner friendly stacks used today as it provides the user with basic knowledge of the technologies that come together. For technologies, a LAMP stack uses Linux as the operating system, Apache for the front-end, MySQL for the database management system and the PHP programming languages for API endpoints. MySQL provides us with a relational database, meaning that we can use primary keys within the database to show a shared relation between two data points. So, for our instance, if we store images and data from our facial recognition software, we can apply those data points to a registered user on the web application. However, a downside to this database is how inefficient it is in terms of replication and sharing, which can cause issues with how often we make a call to retrieve data for our facial recognition software.

MERN stacks are one of the most popular stacks used today for mobile and web development. For Technologies, a MERN stack uses MongoDB for the database. Express and node.js for the backend and React for front end development. Unlike the LAMP stack where we have learned learn multiple languages such as HTML, PHP, CSS, and JavaScript, the benefit to this stack is the fact that everything is programmed using just the JavaScript language. MongoDB is a non-relational database, which acts like a storage system, making it easier to expand and scale our database if needed.

For our project we are choosing to do a MERN stack, it will be cheaper to host using Heroku services and makes mobile app development easier. MERN stack was also selected due to the fact that the whole stack uses one language - JavaScript. The LAMP stack is too out of date and uses multiple different languages, such as CSS, JavaScript and PHP. So, taking the time to learn just JavaScript compared to those three languages will save us time down the road to focus on more development for our facial recognition system.

3.2.1.3 Mobile Application Development

Mobile app development depends on different types of technologies we are going to be using. For our case, we are using the MERN stack. When developing a mobile application, it is important that the app is created for a specific operating system, for our project we believe Android OS is one of the easiest to develop, due to the cost and large market. The MERN stack will allow for easier mobile deployment than the lamp stack and allow us to be more flexible with our web/mobile application overall.

The development of the mobile app will be mostly constructed using the react framework of the MERN stack. We plan on creating an easy to use, straight to the point application where administrators/executives have the capabilities to add and remove Bluetooth tags for guests and employees and have access to view real time footage from the camera. Those users who are not admins will have access to update the tags they are given in case the tags are even broken and they need to request a new one.

Another feature that all common users will have been the availability to scan their face, this data will be used for the training module of our facial recognition software.

3.2.2 Hardware Technologies

In this subsection we describe the background research of indoor tracking and the important components to consider. The team has reviewed and taken this information in before choosing what components we are going to use; it is important to research to see the different options we have out there to make indoor tracking possible to implement for our project.

3.2.2.1 Indoor Positioning Systems (IPS)

Creating an indoor tracking system has lots of benefits in terms of security, and it seems like almost all indoor tracking systems use a network of devices to locate specific objects within a building - in our case individual people. Implementing our system will help establish security measures and liability issues. Technologies such as Wi-Fi, Bluetooth, Ultra-Wide Band (UWB), and Infrared light, have been implemented for various applications of IPS [10].

After considering each solution the conclusion is that Bluetooth (specifically BLE) is a relatively low-cost solution that provides us with a well-documented medium for wireless communication as well as hardware that leaves a relatively small footprint due to its small tags and beacons.

3.2.2.2 Bluetooth Protocols

This project is based heavily on the Bluetooth platform and for this specific implementation we have chosen to make use of Bluetooth Low Energy (BLE) technology. The main benefit BLE has over standard Bluetooth is that as the name suggests it consumes power at a much lower rate by actually staying in a powered down state and occasionally “advertises” to search for compatible devices. The BLE tag in our project will be used for transmitting positional data only when the tag is in range, so there will be a significant amount of time where the tag will not need to operate. BLEs idle mode provides more efficiency over standard Bluetooth implementation.

We have researched multiple Bluetooth protocols, mainly Bluetooth 4.2, Bluetooth 5.1, and Bluetooth 5.2. Bluetooth 4.2 is the original protocol that we planned to include due to its usage in John Teel’s project in which he created a PCB schematic for a basic BLE tag. However, the vastly improved features that Bluetooth 5.2 offers, led us to use this as the Bluetooth protocol for the BLE tags. One such feature is Angle of Arrival (Aoba) and what this does is provides accurate data on not only the distance of a BLE tag (like previous versions) but also the 2D and even 3D position of the tag relative to Bluetooth antennas.

3.2.2.3 IPS Methods

As discussed above, BLE is the chosen technology that will be used to transmit the position of the user. There are many well documented ways of determining a user’s position such as Triangulation, Trilateration, RSSI, and Bluetooth Direction Finding.

The most basic form of distance calculation of Bluetooth is through the use of RSSI or “Received Signal Strength Indicator**”.** Algorithms are created based on the signal strength of the Bluetooth low energy tag and this estimated distance’s accuracy is based on the relative location of the beacon from the tag so is less accurate as the tag is farther from a beacon. While this solution alone is good for estimates of distance with 0.5-2 meters of accuracy, it does not actually determine the angle at which the tag is located just how far a way it is from the antenna. If this method was to be used it would need to be in conjunction with other methods.

There are many algorithms in tracking systems that use this method to improve real time position estimation. With the use of three reference points, triangulation generates an area that considers the center of all three points measured by the reference points latitudes and longitudes.

The next method considered is Trilateration which is the determination of a point in space using three distances measured from anchored antennas. This mathematical method would be used in conjunction with RSSI to give not only distance, but also position of a user.

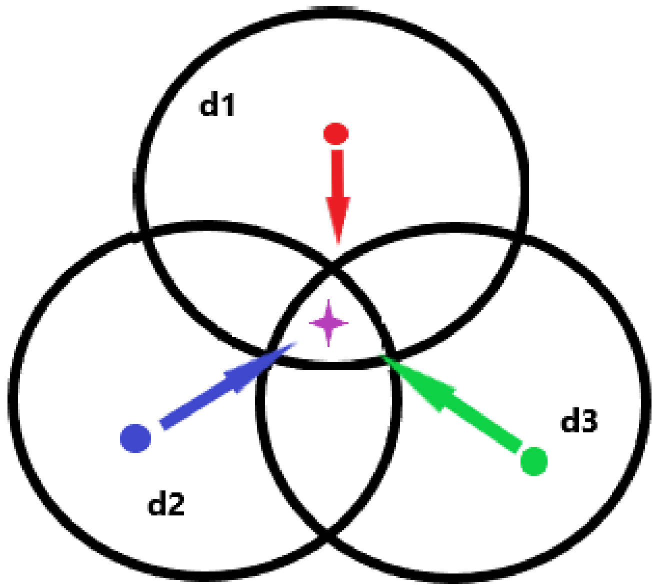


Figure 6: Trilateration Example

Finally, Bluetooth 5.2 introduced the Angle of Arrival (AoA) and Angle of Departure (AoD) capabilities for BLE tags and beacons. This method would allow us to achieve levels of accuracy similar to that of an Ultra-Wide Band solution, while still keeping the low costs of BLE. Bluetooth Direction Finding uses both AoA and AoD to determine the location of a BLE tag based on the phase shifts of the transmitted and received signals. Implementation using AoA allows for the use of simpler tags, while the antennas will be slightly more complicated, whereas AoD will make use of more complicated tags but provide more use towards applications such as wearables, and wayfinding in large areas.

3.2.3 Security System Internet of Things

Being that our project deals with collecting data through sensors, we established a use of an IoT (Internet of Things) approach to keep our devices connected. This approach takes place when a network or system has sensors on one end connecting data and requires that data to be processed by a connected, but separate device in the system.

In our case the sensors will be the BLE tags and beacons that provide location data, as well as the camera that will be running our facial recognition programs for user identification. The camera system requires the position of the tags so that it can adjust the servo motors correctly to target the proper target. And the software application needs all of the data connected so that it can be processed and turned into tangible information, such as displaying the location of a user in the room, and also knowing which person has been tracked as well as showing the video livestream. This wireless connection will be made utilizing a cloud computing solution. The data collected will be stored on this cloud platform and then from there sent to the devices that required the data collected.

The following Diagram shows the general IoT approach this group will be taking in this project:

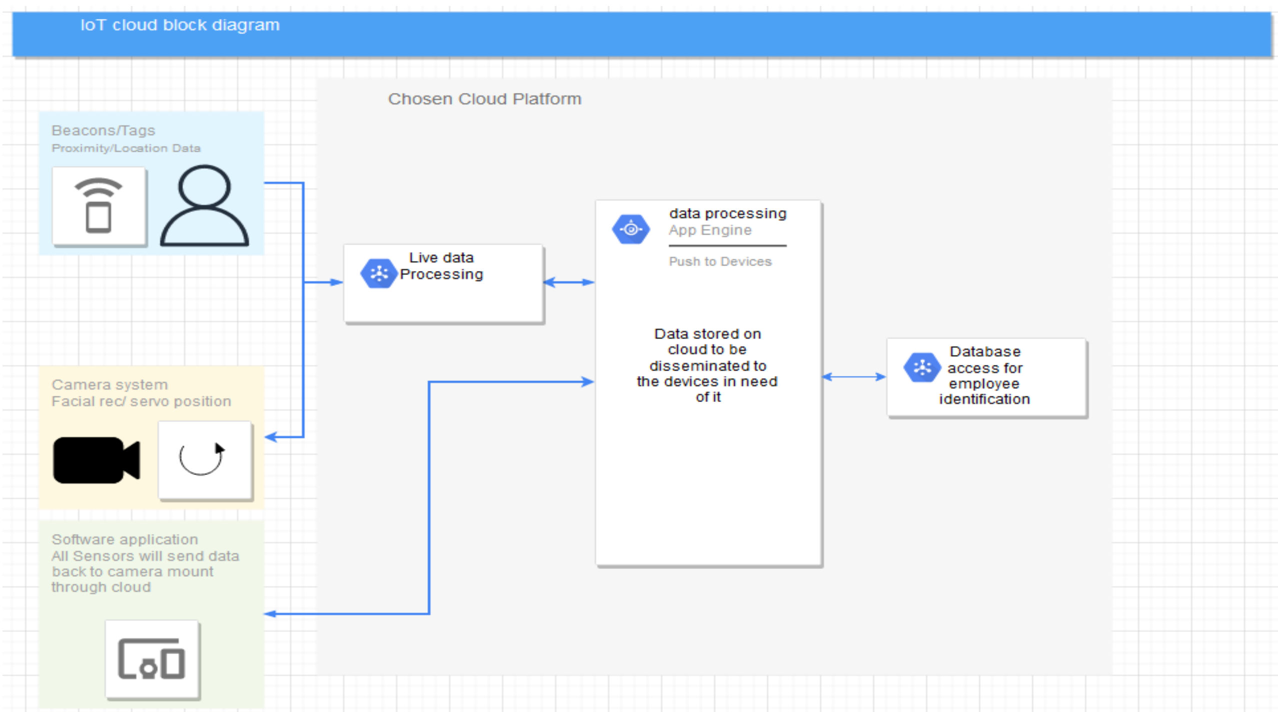


Figure 7: IoT Cloud Block Diagram

Cloud Software approaches

NextCloud

NextCloud is a free and open-source client-server software. This is a cloud software that is favorable for hobbyists and small-scale applications due to it being free as well as relatively simple to install and begin operations on a local server system. NextCloud also provides tools for developers that will allow us to store and display data from any external sensors onto the software platform. This included functionality will allow us to store position data gathered from the Bluetooth tags and beacons on the cloud and will let that data be sent to the application. This reading and writing will be done automatically through the use of Python scripts that will initiate this transmission.

This cloud software can be hosted on a laptop or to keep the design consistent could be implemented on a microcomputer (brand will be based on availability). While NextCloud offers hardware solutions to self-hosting, to keep costs down we have decided not to take this route and move forward with using hardware such as an Arduino, Raspberry Pi, etc. This implementation will follow the format specified above, such that the sensors providing the data will be the same. One downside to this implementation is that we may be restricted by our choice of hardware. If we choose a cheaper MCU/computer, then the read/write speeds can affect the rate at which the live data is processed.

Google Cloud IoT core

Google Cloud IoT core is a cloud platform provided by google to allow for the implementation of end-to-end IoT systems. Google Cloud gives developers access to data analytics tools that help with processing and categorizing data from end-point sensors. This is a great benefit when compared to NextCloud’s platform because we would have access to features that will process certain data for us without us needing to add more scripts/code to the project. Also, unlike NextCloud this IoT core for Google Cloud was made with the implementation of end devices in mind, so live data management and control is made easier when google will do these things for us. Another benefit is that Google Cloud includes an AI that can be used for training and deploying Machine Learning models, which make up a sizable part of our project.

Just like NextCloud Google is not just for enterprises and can be scaled down to be used by a small group of developers. Google Cloud is not open source like NextCloud, however. It is free for a monthly data volume between 0 & 250MB and runs a price of $0.0045 per MB for data volumes between 250MB and 250GB. This pricing range is most likely as high as we go considering the fact that we have not had this cloud process running constantly. As we have only be using it for testing, these cloud prices are acceptable for our group.

AWS (Amazon Web Services) IoT Core

Amazon Web Services is the most widely used cloud platform in the world and we are looking at ways that it could contribute towards our project here. More specifically we have look at the IoT services they offer. Seeing as this service is paid, we have compared it to Google and Azure to weigh the costs and benefits fairly. The Amazon Web Service IoT core is very well documented and widely used so there are a plethora of resources to reference when it comes to implementing this particular IoT project. Similarly, to Google, Amazon Web Service provides data analysis automatically and provides a user-friendly platform for working with data. Through the AWS console we are provided an interface in which we can control the flow of the data collected by our sensors.

The pricing of the AWS IoT core is similar to that of Google, so in our case it seems that our low rate of data usage will result in our group making use of the free tier of AWS. Using this method would allow our group to forgo the use of an extra piece of hardware that would act as our data server. The AI that Amazon provides will also serve a similar purpose in its usage in our facial recognition applications. What Amazon Web Service gives us over Google IoT is the fact that there are so many more resources available on the internet surrounding the use of AWS in smaller personal projects similar to our scale of project. It's for that reason that we value AWS cloud service over that of Google.

Microsoft Azure IoT Suite

Microsoft Azure is another cloud service provided by Microsoft and for our project we focused on Azure's IoT Suite. Azure comes with pre-built IoT solutions which can accelerate the development of our project. The creation of a local cloud is feasible, but if that can be avoided by utilizing Azure IoT then that is the direction the group will take. We found that one of the greatest aspects of Azure is the fact that it is so user friendly. The way that it displays sensor Data on the Azure Hub is very readable and easy to navigate. With our estimated data size per month, we expect to be using the free tier of Azure IoT Hub as well. This price is acceptable considering our project is self-funded.

A unique benefit of using an Azure IoT solution is that Microsoft is supportive and encourages the use of open-source Software Development Kits. This makes the use of our Jetson nano and microcontrollers a bit easier since we can see open-source projects in which similar devices are utilized by other developers. Microsoft allows us to develop in many languages in order to develop communication protocols. Overall, Azure provides a plethora of resources and gives customization options that the other cloud platforms just don't have in the same capacity. As the pricing is also within our group's price range Azure is a top choice for us.

Local "Cloud" approach

This is the generally much cheaper option, in which our group will create a cloud-like service locally and use it to store data that can be accessed by the machines in our network. We can use a remote MCU or Microcontroller as our "cloud server" and on this central device we communicate wirelessly from all edge devices to send and receive data.

This approach gives us the most control over how our devices will communicate wirelessly and allows us to avoid having to pay to use the cloud on a per/minute or per/hour cost as many of the above solutions would have us do. A downside is that since this will be a novel setup the option to display our data will not be the same as other cloud solutions, but this can be avoided by representing our data in a way that no manipulation will need to be done manually by us during runtime of the project.

This communication can be done through Wi-Fi and will make use of socket programming to send and receive data as we need it. This requires the creation of our own data organization standards as well as keeping a common protocol among all of our edge devices. It is important to keep the speed of this communication in mind and considering the application we need a device that allows for Full-Duplex communication to keep the real-time system running at maximum efficiency.

Mosquitto MQTT Broker

Mosquitto is an open-source message broker that is commonly used for communications between a microcontroller and web applications. Mosquitto implements the MQTT protocol which provides a lightweight method of carrying out messaging using a publish/subscribe model. This broker is suitable for various Internet of Things messaging [40].

A unique benefit to this broker is that it is popularly used with the ESP module which is what our group is leaning on using. Its high speed and wide variety of implementation with different languages make this a prime contender for our communication between our microcontroller and application.

Mosquitto will be responsible for creating a server on our local host, at this point our microcontroller will be programmed to connect to our server and to subscribe to a topic, in which it will send over messages to the broker. Once the broker receives the message, he will publish it on the software application which will similarly have to connect to the server and the same topic to receive the message. Ultimately, the MQTT protocol allows for reliable messaging between devices, has security capabilities and allows for Bi-directional communications.

Socket.IO

Socket.io is an event-based communication protocol. Socket.io implements the WebSocket communication protocol which provides a full-duplex and low latency channel between the server and the browser [41]. Sockets are commonly used when wanting to constantly get real time data, so in our case either our tag and beacon system or our face recognition system.

Although Socket.IO is not best suited for IoT devices, it can be useful when communicating between devices. We are learning to use Socket.io to send real time data from our face recognition system to our software application. Overall, sockets are very popular for communication channels utilizing its websocket protocol.

3.3 Strategic Components and Part Selections

This section will go into detail about the major parts of the Indoor BLE Security system as well as the accompanying research with each component. You will see various diagrams and figures that help explain each component and how it will be used for each of its corresponding systems. The information seen in this section was used to make the final selection of the parts that will be utilized in our system.

3.3.1 Communication Devices

Indoor locational tracking is a major component. Different communication protocols exist to make indoor positioning possible. In this section we discuss different communication systems that were considered and chosen for making indoor tracking possible.

3.3.1.1 Radio-Frequency Identification

Originally for indoor locational tracking, RFID was a top contender. Radio Frequency Identification or RFID was the first option we looked at. RFID “is a technology that uses radio waves to make a specialized circuit produce a response containing a unique identifier,” and consist of RFID readers and RFID tags. The readers attempt to pick up the data emitted from the tags.

RFID are used in various applications, but due to its limited signal capabilities for indoor location tracking it has been used for localization proximity. This means that instead of displaying or emitting locational data in a coordinate system of X, Y, Z, the data is given in the form of a logical location such as “Inside the waiting room”, and “Inside room 123.” However, one issue that arises is the bleed through of the signal the tags can emit. Since the radio signals can essentially go through walls, it can activate several readers within the area and activate all of them, overall flawing the indoor positioning system.

3.3.1.2 Bluetooth Low Energy

Bluetooth low energy is one of our top options for implementing our indoor positioning system. Bluetooth is “wireless communication technology that uses digitally embedded information on radio frequency signals,” [8] and focuses on facilitating communication between devices, in this case a Bluetooth beacon and a Bluetooth tag. Bluetooth low energy has relatively low cost and low power consumption compared to its competitors in this section.

Bluetooth low energy is used in various applications of localization, its most popular configuration involves having a fixed beacon with a tag that emits small radio waves, overall increasing the accuracy of the system. Similarly, to RFID it can be used as a proximity locator which can identify if a person enters a specific room, or we can use it to transmit location data in the form of the coordinate system depending on which multilateration algorithm we decide to use. It is possible to make the location position visual by creating an indoor map of the space, but for now we want to focus on having our system detect location proximity. From our understanding, there are not really any underlying issues to watch out for, of course BLE is not perfect but compared to its counterparts of communication devices, it is near the top due to its accuracy, and low-cost system.

3.3.1.3 Wi-Fi

Wi-Fi is another premier option that was considered for the indoor positioning system. Wi-Fi "transmit and receive data using electromagnetic waves, providing wireless connectivity within a coverage area" [15], meaning that if you are within the area of coverage for Wi-Fi you will be able to communicate through the network. Wi-Fi communicates through an access point and typically a smart device or tag that accompanies the indoor tracking system.

Typical Wi-Fi based systems provide an accuracy of 3-5 meters, which is less than suitable for our intended solution to this IPS problem. Wi-fi systems work by sending packets to some Wi-Fi compatible device and using established algorithms to determine distance based on packet transfer time and signal strength. By using a RSSI, and a MAC address we can calculate the current location of the end user device within the area. One of the downsides to this implementation is the price of the Wi-Fi tags are much more expensive than the BLE tags that are mentioned above.

3.3.1.4 Ultra-Wide Band

Ultra-Wide Band (UWB) is "based on the transmission of electromagnetic waves formed by a sequence of very short pulses using very big bandwidth," [14] offering precision and multipath immunity. Similarly, to Wi-Fi and Bluetooth, UWB also used a system of beacons/access points and tags to implement the indoor tracking system.

Ultra-Wide Band (UWB) provides almost up to 30cm of accuracy, which is excellent, but the installation of these systems can be very expensive (can be $150 per anchor/beacon) and is very complex to implement. Overall, UWB is one of the most popular choices when implementing indoor tracking due to its speed and precision.

3.3.1.5 Visible Light

We also considered Infrared sensors for transmission of positional data. However, the receiver modules for IR IPS systems require well established infrastructure that our solution doesn't meet and is often used in tandem with other methods of IPS to cover deficiencies.

3.3.1.6 Graphical Positioning System

Graphical Positioning System, better known as GPS, was initially a solution to finding real time location with precision. After some research, the practicality of implementing GPS indoors became less apparent. Though it is used widely for tracking location for many devices, GPS would not be a suitable option for indoor tracking. The materials used to maintain the structure of a building blocks the signal between the transmission device and the receiver. This makes real time tracking indoors impossible to implement utilizing the real time capabilities of using a Graphical Positioning System.

3.3.1.7 Wireless Communication Selection

Table 1 below shows our top 3 options that were considered for our indoor positioning system implementation. You will see that we removed the RFID, Visible Light and Graphical Positioning systems. Visible light and GPS systems were removed pretty quickly, due to problems described above. Using RFID handheld devices such as tags and RFID readers was overall too simplistic for our idea of indoor tracking and not feasible due to the lack of the technology protocols overall. Which leads us to look at the remaining communication devices below.

Type	BLE	Wi-Fi	UWB
Location Accuracy	< 5 m	< 10 m	10-50 cm
Range	Up to 100 m	Up to 500m	Up to 200m
Latency	3-5 seconds	3-5 seconds	1 ms
Power Consumption	Low	Moderate	Low
Frequencies	2.4 GHz	5 GHz	10.6 GHz
Data Rate	2 Mbps	1 GBps	27 Mbps
Cost	$50-$75	$150-$200	$70 - $100

Table 1: Wireless Communication Comparison

The three options for communication protocols listed above are all very popular when it comes to the implementation of the indoor positioning system. For our case, the team decided to go with the BLE technology. UWB was close to being selected due to its accuracy and latency, however the implementation of UWB was too complex for our liking and due to time considerations with time constraints we decided that it would be best to look at either Wi-Fi or BLE. Wi-Fi was a great choice as well but due to its high cost and its high-power consumption we ultimately ended up choosing BLE. BLE provided us the best option in terms of cost and accuracy, also compared to its other counterpart, easier system to implement. BLE also allows us to not only do locational proximity tracking, which is what the group first plans on creating, but if possible and if time allows it, we can create a visualization location system to see the tracking come to life. Now that we have chosen our communication system for positioning tracking, we must pick which beacons and tags we are going to use, which will be discussed in the next section.

3.3.2 Bluetooth Hardware

Bluetooth low energy is a very common indoor tracking system. BLE uses a system of beacons and tags to emit location data. Basically, we have beacons set up in rooms, when a tag comes within the proximity of the beacons, it will emit a signal to the nearest beacons to show where it is relative to the beacon. In this section We have go over Beacon transmission concepts, as well as which beacon and tag system our team will use for this project.

3.3.2.1 Beacons Transmission Concept

A Bluetooth beacon is a “small wireless device that works based on BLE.” The beacon acts as a lighthouse to its corresponding tag, and broadcasts radio signals. Each beacon is made from a CPU, radio and batteries. Since it is constantly emitting radio signals, the Bluetooth tag picks up the signal, once it is connected it will carry out whatever function it has been programmed to perform, in our case location tracking of the tag. Below we discussed various transmission concepts popular with BLE beacons.

3.3.2.1.1 Far Field Region

Field strength decreases as the distance from the source of transmission increases. The power intensity of electromagnetic radiation will experience the inverse-square law.

3.3.2.1.2 Near Field Region

Field strength increases as the distance from the source of transmission decreases. The power intensity of electromagnetic radiation will experience the inverse-square law according to everything [38].

3.3.2.1.3 PCB Ground Plane

The way that a ground plane works is it works as an extension to the length of the antenna. When the ground plane is considered in the equation for signal strength, it extends the signal by multiple/ratio of the total combined length (antenna length + ground plane length).

3.3.2.1.4 Radiation

When an Antenna transmits an electromagnetic signal, this process is called radiation. Antenna's signal pattern will directly associate with the strength of the signal and the antenna orientation.

3.3.2.1.5 Directional Antenna

Directional antennas work in situations with interference; however, the signal strength is negatively impacted by metal. In reference to tuning the antennas, *** plastic, glass, or ceramics throw off the resonant frequency of the antenna ***

3.3.2.1.6 Omnidirectional Antenna

Omnidirectional antennas do not work in situations with many interferences. Human skin would be an interference to the signal being transmitted from a Bluetooth device such as a smartwatch or wireless headphones.

3.3.2.1.7 iBeacons

Developed by Apple, these Bluetooth enabled beacons transmit low energy signals to other Apple devices by sending a Universally Unique Identifier, better known as UUI or UUID in some cases. For lasting life, iBeacons are powered using batteries typically spanning a life of two to four years on average.

3.3.2.1.8 Eddystone

Developed by Google, these Bluetooth enabled beacons transmit low energy signals to devices supporting both iOS and Android by sending a UID, EID, TLM, or URL. UID denotes Universal Identifier, EID denotes Encrypted rotating Identifier, TLM denotes Telemetry, and URL denotes Uniform Resource Locator. For lasting life, Eddystone beacons are powered by batteries spanning about four years on average.

3.3.2.1.9 Altbeacon

Developed by Radius Network, these Bluetooth transmit low energy signals across various platforms, similar to iBeacons it uses an UUID. Altbeacons are often used to define specifications for device transmission and have tools suitable for development applications. Similar to iBeacons they are powered by batteries lasting about two to four years on average

3.3.2.2 Beacons

As stated above, Beacons are used to transmit signals at set intervals so that tags within the proximity will be able to listen to these signals and react accordingly. This section will cover the different beacons our team looked at and which one was selected to be used for our project and the criteria we used to select this component.

	Sticker Beacon	Proximity Tag	Proximity Beacon	Location Beacon	Location UWB Beacon	Video Beacon	LTE Beacon
Built-in communications	Bluetooth 4.2	Bluetooth	Bluetooth 5.0	Bluetooth 5.0	Bluetooth 5.0 UWB	Bluetooth 5.0 Wi-Fi	Bluetooth 5.0 GPS
Default battery life	1 year	2 years	3 years	4 years	3 years	USB	2 years
Maximum Battery Life	1 year	3 years	5 years	5 years	5 years	USB	USB-C
Maximum Bluetooth Range	7 meters	70 meters	100 meters	150 meters	200 meters	10 meters	200 meters
Supported use-cases	Asset Tracking	Proximity asset tracking	proximity	Indoor location	Indoor location	Digital signage	Asset tracking, vehicle tracking
Thickness Length	6mm	15mm	25mm	24mm	27mm	14mm	16mm
Width	28-44mm	38mm	46mm	41mm	50mm	60mm	90mm
Weight	8-12g	30g	86g	67g	98g	72g	57g
Recomme	Estimote	Proximity	Proximity	Indoor	Indoor	Mirror	Web IDE

nded SDK	SDK	SDK	SDK	SDK	SDK	SDK	
Fleet Managem ent	Estimote App	Estimote SDK	Estimote SDK	Estimote SDK	Estimote SDK	Cloud via Wi-Fi	Cloud via LTE

Table 2: Comparison of Beacons

3.3.2.2.1 IP67 Waterproof Bluetooth Beacon

The IP67 Waterproof Bluetooth Beacon is a BLE beacon that runs Bluetooth 5.1 and above. This beacon supports iBeacon, Eddystone and Altbeacon broadcasting. This beacon has a long battery life of about four to six years and is 48mm x 37mm x 7.8mm in size, so it will be easy to place anywhere we choose to. This beacon comes pre-programmed with Feasycom standard Beacon firmware making its application easy to use to implement Proximity Tracking.

3.3.2.2.2 BLE Beacon 5.0 Low Power Consumption

The BLE Beacon 5.0 Low Power Consumption is a low cost BLE beacon which runs Bluetooth low energy at 5.0 protocols. This beacon supports only the iBeacon platform. This beacon has a lifespan of about two years and is 34mm x 30mm x 8.4mm in size. This beacon is primarily used for indoor location applications and can be easily placed anywhere in a room location due to adhesive stick-on feature.

3.3.2.2.3 Dialog DA14531 BLE Beacon

The Dialog DA14531 BLE Beacon is a BLE beacon that supports Bluetooth 5.1 and above. This beacon supports iBeacon and Eddystone platforms and has a long battery life of about eight to ten years. The beacon is 86mm x 65.7mm x 17.2mm in size which is bigger than the previous two but emits a stronger signal, meaning that we need less beacons to cover an area of the room.

3.3.2.2.4 Beacon Selection

Table 3 below shows the comparison between all three beacon options and the final selection of the beacon being used in the table you will see all the different comparisons between each beacon system. The highlighted column is the beacon we have selected

Name	Dialog DA14531	BLE Beacon 5.0 Low Power	IP67 Waterproof

	BLE Beacon	Consumption	Bluetooth Beacon
Platform	Eddystone / iBeacon	iBeacon	Eddystone / iBeacon / Altbeacon
Dimensions	88 x 65.7 x 17.2mm	34 x 30 x 8.4mm	48 x 37 x 7.8mm
Net Weight	28g	6.7g	15g
Operation Voltage	1.8 - 3.6v	1.7 - 3.6v	1.7 - 3.6v
Transmission Range	400m	50m	400m
Transmission Power	-30dB to +4dB	-40dB to +4dB	-40dB to +4d
Output Power	+4dB	+4dB	+4dB
Operating Temperature	-40°C to +105°C	-20°C to 70°C	-20°C to 70°C
Broadcast Interval	100 - 2000ms	100 - 1000ms	100 - 1000ms
UUID / Major / Minor	Static	Programmable	Programmable
Working Time	8 to 10 years	~2 years	~6 years
Price	$10.00	$7.00	$12.00

Table 3: Beacon Selection comparison

The three beacons above are all fairly similar, in terms of size and operating voltages. We decided to go with the Dialog DA14531 BLE Beacon due to its cheap price and high transmission range that it offers, which gives us more flexibility when locating our beacons in our test area for our prototype. Although we get a higher broadcast interval which is longer between ping, we felt like that will not make a major difference in our system and that everything should work up to our standards. For these reasons we selected the DA14531 BLE beacon, and our next step is to get a corresponding tag to come one step closer to completing our system.

3.3.2.3 Tags

Tags go hand in hand with the BLE beacons. Once the beacons have been set up in the room location, individuals will be given a tag to carry. The beacons will be constantly emitted Bluetooth signals when turned on, the tags are ready to receive the signals. Once the tags come within the signal range of the beacons, the beacon will notify the system that the tag is in the proximity emitting a message that states - “tag number two is in room one.” Tags can be in the form of phones, watches, etc., however for our case, our team decided to go for an asset tag card, similar to ones that employees for major businesses have.

3.3.2.3.1 Tag Selection - Dialog DA14531 Asset Tag Card

Once we selected the beacon, selecting the tag was easy. The Dialog DA14531 BLE Beacon has a corresponding tag that can be used alongside it. The tag is usually used for applications such as asset tracking, indoor location and proximity location, which is perfect for our project. It has a transmission power of about -20dBm to +4dBm and runs version 5.1 and above of Bluetooth. This tag also comes with an NFC feature which allows us to add extra security measures for our system in the future. Ordering this is to test out our original beacon and its functionality, but we plan on using the microcontroller to be our final Bluetooth tag as it would be easier to transmit the data to our IoT over UART communication.

3.3.2.4 Battery

Now that we have outlined the technologies necessary to create a network of wireless devices, it is imperative to also explain what drives these devices. Power is a major component to any project that requires technology. The system that we plan to execute will incorporate more than one form of power. To start, the microcomputer will be powered by a wall socket source. The wall socket will provide an alternating current source which allows the voltage to be significantly higher than a power source that operates on a battery of some fashion. According to NewbieHack [39], the voltage range provided by the wall power is from 110-130 volts. The benefit of having such a high voltage is having the ability to provide power for multiple devices simultaneously. Another benefit of having a wall socket is power reliability. Barring out power outages and damaged sockets, wall socket power will never need to be changed out. In the project, the microcomputer will need enough power to control connected peripheral devices so there is no worry of needing too much power for the project to function properly. This is where another power source comes in. USB is used to supply power to one device and provide direct current at around 5 volts. 5 volts is more than enough power for the servo motors to receive to function as expected. The servos are not the only peripheral needing power managed by the microcomputer. The camera module will also need power to both turn on and run the algorithms that are driving the face detection software.

There are two possible implementations which impact what battery source will be used in the indoor positioning tracking system. In the first option, there will be a single MCU and at least 3 BLE Beacons. To power the MCU, the potential options are wall socket, USB, or even battery pack. Though it would be functional from the position of being plugged into a wall, the mobility of the MCU will be limited by needing a wire hooked up to an outlet for power. For cost efficiency and effectiveness, USB is the best option for the MCU. In the demo implementation, a laptop will serve as a mobile power supply via USB. This way our indoor positioning system does what it was intended to do. This transitions into the next source of power which is great for the beacons but a questionable option for the MCU in this implementation of the IPS. The beacons should not take up too much space nor should it require more than 3.3 volts to operate for at least 1-2 years. Using CR2032 batteries will give us the flexibility to do both whereas the wall socket and USB options will not. Being compact is important because larger spaces which will require more beacons (3 beacons will be used for the demonstration) will need to be easily accessible but also out of sight.

In the second potential IPS solution, there would be 3 MCUs and a single beacon. In this situation the roles are different. In the previous option, the beacons will be using their signals along with a trilateration algorithm to approximate the real time location of the MCU. The second solution will just do the same thing in reverse, but the power options will be different for the MCUS for both the demonstration and the real-world application of this prototype. The beacon will still use a 3.3-volt coin battery. The MCUs will need a different approach than carrying around 3 laptops and powering them via USB. For the demonstration, this would work only to show that the indoor positioning system is accurately measuring the beacon in relation to the other MCUs. To maximize the usefulness of the MCU, rechargeable batteries would best suit the prototype. Before addressing the benefits of integrating rechargeable batteries into powering the MCU, it would help to look at why the other options are not as feasible. Wall socket is out of the question because each MCU would require a wall socket and the voltage provided by the wall is unnecessarily high for the MCU to function. USB will provide the power needed to broadcast the signal; However, providing power to the USB via a laptop would not work in a business setting where these beacons need to remain compact and unseen. Hooking a battery up is the most approachable solution although there are a few cons here as well. Using a non-rechargeable battery introduces the issue of decreasing the projected maintenance from 1-2 years (previous implementation) down to somewhere around 1-3 months of use time before needing to be switched out which ultimately drives up the overall cost to a potential investor. This is why rechargeable batteries will fit so well here. Though the rechargeable will be more expensive than the non-rechargeable, the benefits outweigh the difference in cost. Instead of replacing the battery every 1-3 months, the battery would need to be charged once a month for the first year. After the first year, the battery will need to be charged 1-2 times per month until the 2-year mark hits and the battery becomes too depleted to continue recharging it.

3.3.3 Microcontrollers

By definition, a microcontroller is an integrated Circuit (IC) that is programmed to perform a specific task. Components on a microcontroller consist of a CPU, I/O Ports, RAM, Oscillators, and Power Supply amongst many other components not listed. Microcontrollers have few parts in comparison to microprocessors that can host a lot of parts and power them.

The Central Processing Unit is the brains of the microcontroller. This part sends instructions between the main memory and the Arithmetic Logic Unit (ALU). The ALU's job is to crunch the numbers and the operations and send the results of the operations to the main memory. The memory of the MCU is split into two types, program memory and data memory. Program memory is non-volatile memory that stores long-term information received from the CPU and it holds this information even when there is no power passing through the microcontroller. Data memory is volatile memory that stores temporary instructions received from the CPU; However, these instructions get maintained only when power is present within the microcontroller. The Control Unit (CU) will fetch the instruction from the memory and send out the data. This data is sent using the Input / Output peripherals (I/O) attached to the microcontroller. The goal of these interactions within the microcontroller is to perform a specific function for a larger system.

Random Access Memory, better known as RAM, is a type of volatile memory used in microcontrollers and alongside processors in larger computational devices. Erasable programmable read-only memory (EPROM) and electrically erasable programmable read-only memory is both non-volatile memory types.

The serial bus interface will play a significant role in sending programs for the microcontroller to fulfill. The Universal Serial Bus (USB) will be used as the serial bus interface for transferring either data or power to the MCU. It is also important to mention that the microcomputer will also use the serial bus interface to either transfer data or provide power to peripherals.

The Integrated Development Environment (IDE) will be used to program the microcontroller. Microcontrollers do not typically have the same memory as a microprocessor. In fact, most microcontrollers only have around two hundred and fifty-six bytes which is just two hundred and fifty-six registers holding eight bits each. Direct access to these bits typically requires a high-level programming language. These languages are C, ASSEMBLY, Python, and Java. Within the compiler of the IDE, the instructions are placed onto the memory

Analog to Digital Converter (ADC) will take samples of waveform and measure the voltage fluctuations over a time frame giving us the frequency. This information on the microcontroller will give us the data for voltage output. Digital to Analog Converter (DAC) will take a digital value and convert it into a voltage.

3.3.3.1 Microcontroller Selection Criteria

Currently, there are a bunch of great options for microcontrollers out there and it’s hard to compare them with each other because some of them are designed for different purposes. In this section you will read about the different criteria considered for selecting out top three options of MCU’s.

3.3.3.1.2 Operating Voltage

The operating voltage gives a range of voltages that the microcontroller will need to function as intended by the manufacturer.

3.3.3.1.3 Temperature Range

The board has both lower and upper bounds for operating temperature. Once the temperature reaches outside the marked temperature limits, the chip will begin to throttle, and operations will slow down as a result of this.

3.3.3.1.4 Max Clock Frequency

The integrated circuit takes in a clock input which has a certain operating frequency. The highest frequency the chip can operate at is denoted by max clock frequency.

3.3.3.1.5 Flash Memory

Non-volatile memory that is used to store application program code and data. The memory cannot be overwritten while a program is executing because it has read-only permission during this time.

3.3.3.1.6 RAM (Random Access Memory)

Volatile memory that is used to store program data. In comparison to Flash Memory, RAM has less latency making it more expensive than Flash. Generally, microcontrollers will have more Flash Memory available to utilize than RAM.

3.3.3.1.7 Analog I/O

The Analog signals are those that account for measurements that impact the functionality of the device. Analog input signals typically transmit from a sensor which is then converted into a digital signal by the ADC. In conjunction, the Analog output signals typically travel from the computer to a measurement apparatus but before it does so, the signal must go through the DAC.

3.3.3.1.8 Digital I/O

This component will transmit a digital signal in either a low state or a high state. When the digital input receives voltage, the computer will recognize it as low or high based on a threshold requirement. Digital output is controlled by the computer to produce the low or high state based on the instructions.

3.3.3.1.9 GPIO Pin Count

General Purpose Input/Output pins have two voltage settings, low and high. These are programmed using software.

3.3.3.1.6 Bit Count

Bit count determines the size of each word. This number usually varies from microcontroller to microcontroller. Common sizes are, eight bits, sixteen bits, thirty-two bits, and sixty-four bits.

3.3.3.1.7 Low Power

A feature of Analog devices, lower power mode ensures the operation of processing localized data while preserving the most battery life to extend the life cycle of the battery.

3.3.3.1.8 Power Consumption

Bluetooth low energy technology consumes just that, low energy. The signal is meant to be powerful enough to reach about ten meters with no interference meanwhile draining the battery at a very slow rate.

3.3.3.1.9 Price

Our budget allows us to get a decent MCU based on the requirements of our indoor position tracking system.

3.3.3.2 Microcontroller Comparison

Our project requires specific task control to operate the servos and the camera system to communicate. In this section we go over the top choices we came across and select the best one for our project.

3.3.3.2.1 CC3200

One of the few Wi-Fi and Bluetooth enabled boards available in TI’s vast line up of microcontrollers, the CC3200 is a viable option for our indoor positioning system. This versatile board features the following, 40 pins GPIO, 4KB of RAM, and 128KB of Flash Memory to tap into.

3.3.3.2.2 ESP8266

In the initial research for microcontrollers, this particular device along with the ESP32 ranked highly on more than one website in using a microcontroller for Internet of Things products.

3.3.3.2.3 ESP32

Similar in design and functionality to the ESP8266; However, there were improvements made to this chip that gives it a slight edge in comparison.

3.3.3.3 Microcontroller comparison

The table below features a breakdown of the 3 microcontrollers we looked at in our research, as well as technical specifications that we closely compared to help come to a decision about which MCU we would end up using in our final product.

Board	CC3200	ESP8266	ESP32
Manufacturer	Texas Instruments	Expressif	Expressif
Operating Voltage	2.3 - 3.6v	2.7 - 4.0v	2.6 - 3.8v
Temperature Range	-20°C to 70°C	-20°C to 70°C	-20°C to 70°C
Max Clock Frequency	MHz	MHz	MHz

Flash Memory	128 KB	256 KB	32 KB
RAM	4 KB	8 KB	8 KB
GPIO Pin Count	40	38	32
Bit Count	16	8	8
Low Power	Yes	Yes	Yes
Power Consumption	Low	Low	Low
Price	$55.00	$7.39	$10.00

Table 4: MCU Comparison

3.3.4 Cameras

The camera selection is a major component as we use it to run our facial recognition software. Using computer vision, the camera will be responsible for target detection/tracking of the individual who walks inside within the angle of view. The camera will be responsible for identifying the individual using the facial data stored in the database. For object detection, a camera with good resolution and good frames per second is essential.

3.3.4.1 Raspberry Pi HD Camera

The Raspberry Pi HD Camera module is a high-definition programmable camera sensor. It has a max of 12.3 MP quality, which maps out to be around 4096 x 3000 for the resolution. If we were to use it on high quality, this would drop the frames down to about 15 fps. However, this selection here will allow us to drop to 1080p resolution with a range around 30 fps. From previous research done in the past, this is sufficient for a facial recognition system. A massive plus is the fact that the camera is programmable. This allows us to upload our program to the module itself and retrieve the data to display it on screen, making the module its own entity.

3.3.4.2 Logitech C920

The Logitech C920 camera is a high-definition, mounted webcam. From previous research, we see that this series of cameras is highly popular – one of the most commonly used cameras from past senior design projects. This camera offers us 1080p at 30fps, with a 78-degree field of view. It also has auto-focus which makes it attractable to use for object recognition software. It works by connecting to a computer using an USB 2.0 connection. It will be necessary to have this camera always plugged in to a computer to run the software and to retrieve data. Since it relies on the connection to a computer, implementing this camera would be easy since we would not have to design any specific connection for this sensor.

3.3.4.3 Lenovo 500 FHD

The Lenovo 500 FHD webcam is one of the cheapest and optimal options we found when selecting cameras. Similar to its competition it runs at 1080p at 30 fps. It only provides about 75-degrees of freedom for the field of view. It also comes in with built in 360 degrees pan/tilt controls which is a plus when trying to track objects. The camera uses a USB 2.0 connection, which again makes implementation easier between our facial recognition software and camera. A big plus with this camera is the 4x Zoom feature, which will come in handy when trying to detect objects.

3.3.4.4 Camera Selection

Table 5 below shows the comparison between all three camera options and the final selection of the camera being used. The features we particularly look at include resolution, field of view, and frames per second. These are ultimately the most important features we looked at when considering which camera to use, as we believe they can affect the overall accuracy of our computer vision component. Price was also a contributing factor in our decision, but the technical specifications likely will have a stronger influence on our decision.

Type	**Logitech C920**	**Lenovo 500**	**Raspberry Pi Hd**
Resolution	Full HD 1080p	Full HD 1080p	12.3MP
Field of view	78 degrees	75 degrees + 360 pan/tilt controls	80 degrees

Frames per second	30fps	30 fps	15fps at max, up to 30 fps
Digital Zoom	Yes	Yes	no
Autofocus	Yes	Yes	Yes
Weight	5.71 oz	8.8 oz	7oz
Dimension(inches)	1.7x3.7x2.8	1.98x4.2x2.48	1.7x3.7x2.8
Price	$69.99	$39.99	$69.99

Table 5: Comparing Cameras

The three cameras above are fairly similar, all offering us 1080p with up to 30 frames is pretty standard for decent facial recognition detection. The Lenovo 500 is the cheapest option and gives us everything that the Logitech C920 offers at almost half the price. Our other option was the Raspberry pi HD camera, which provides us with better quality images at a decrease of frames per seconds. The choice came down between the Raspberry Pi HD camera and the Lenovo 500, which differed in price and how we choose to set up our camera, whether we want to run a USB cable for our software or upload it to the camera module itself. The camera is a major component for this project and in security systems in general, usually the cameras are their own entity in the business security world. For this reason, the selection will be the Raspberry Pi HD camera module, to create an independent component of the security system.

3.3.5 Servos

Servos will be used to engage the pan and tilt mechanism of the Camera – to position the camera with horizontal rotation of 180 degrees and adding vertical tilts. The system will be responsible for assisting the camera with detecting facial recognition. The degree of motion available from the servos will affect the efficiency and accuracy of the camera’s facial recognition software we implement.

We plan on programming the servos, as it will allow the camera to get into ideal position to track the user. If we were to use continuous servos, it would constantly be stuck in a 180-degree loop, never fully detecting a user within the camera's field of view. For this project, we have a selection to choose from 180 degrees motion servos and 360-

degree motion servos. Using a 180-degree system will require our servo to handle more torque, which will take more voltage levels. Having a continuous 360-degree servo will require less voltage and will decrease the torque of the system, however it will require the servo to rotate the area for longer, decreasing the system's responsiveness.

3.3.5.1 Arducam Pan Tilt Platform

One of our options is the Arducam Pan Tilt Platform. This is a pre-built platform that comes in with pan and tilt controls. The servos on this build are compatible with raspberry pi, and Nvidia Jetson, which are 2 of our choices when it comes to selecting the MCU. This Pan Tilt Platform has a vertical rotation angle of 180 degrees. For torque, it is very dependent on the voltage level we run through. For example, this platform can create 0.6kg/cm at 3.6 V, or 0.8kg/cm at 4.8 V [5]. The nice feature of this module is the fact that everything comes built in, all we are responsible for is implementing the working code.

3.3.5.2 WaveShare 2-DOF Pan-Tilt HAT

The WaveShare 2-DOF Pan Tilt HAT is another option we have for our servos. Similar to the previous option this platform is also compatible with Raspberry Pi, and the Jetson Nano. This platform provides slightly less torque than the previous system, however it still provides the same vertical rotation angle of 180 degrees. Similar to the last pre-build we are responsible for implementing the code to get the servos to move as wanted.

3.3.5.3 SunFounder 2 Axis Pan Tilt Kit

The SunFounder Pan Tilt kit is an all-in-one servo kit that is popular to go together with Raspberry Pi Camera Module. The platform allows for 180 degrees angle movement and offers 2 - Axis pan tilt for the camera, it is a popular choice for color, gesture and facial detection projects. The kit comes assembled, the only change is applying our camera module to this Pan-Tilt HAT and removing the existing one, which being that the raspberry pi Hd camera is the most updated module, should not be a major difference when replacing the pre-existing 5MB raspberry pi camera. Although the reviews and feedback about this product is not the best, we are open to adding improvements to this existing module, it also helps knowing that the software to help the camera with facial detection is built into the servo CPU.

3.3.5.4 Selection of Servo Platform

Table 6 below shows the comparison between all three camera options and the final selection of the camera being used.

Platform	Arducam Pan Tilt	WaveShare 2-DOF	SunFounder 2

	Platform	Pan-Tilt HAT	Axis Pan Tilt Kit
Dimension	20mm x 8.75mm x 22mm	56.6X65(mm)	23.2 X 12.5 X 22.0 mm
Speed (4.8v)	0.09sec/60°	0.12sec/60°	0.12sec/60°
Torque (4.8v)	0.8 kg/cm	3.5 kg/cm	0.8 kg/cm
Pulse Amplitude	3.6v/4.8v	3.3V/5V	3.3v/4.8v
Price Range	$26.99	$23.99	$29.99

Table 6: Servo Selection

The three servos above all are fairly similar. For this project we personally decided to go with the Arducam Pan Tilt Platform. The Arducam platform is a popular product to use with Raspberry Pi HD cameras, so there are a lot of different resources out there to help us with the implementation of facial recognition to configure with our camera system. Also, this product is heavily popular with Nvidia Jetson Nano, which is one of the choices for MCU our team is deciding upon. Also, the price is a good in-between given our different options/selections.

3.3.6 Microcomputers

For our project a microcomputer is needed to create and deliver facial recognition code and camera information to our security system. In the search for the most appropriate micro board size, weight, and functionalities were taken into account the most. To control Servos, the chosen microcontrollers must be able to send a PWM signal to interact with the servos for positioning. As mentioned before the computer must be capable of running machine learning algorithms in order to train our facial recognition software. The following microcomputers were the ones that were reviewed in the decision-making process and their strengths/weaknesses and specs are shown as well.

3.3.6.1 Raspberry Pi Model 3B+

The raspberry pi 3B is a microcomputer that provides 1GB of RAM and is capable of Bluetooth 4.2 wireless communication as well as Wi-Fi. The model 3B+ was the initial microcomputer that we evaluated, but it doesn't provide the computing power that a project involving machine learning and facial recognition will need with its limited processing power.

3.3.6.2 Raspberry Pi 4 Model B (4GB & 8GB)

The model 4 provides a significant upgrade to its predecessor the model 3B+ with its 4 or 6 Gigabytes of RAM, and support for Bluetooth 5.0. Raspberry pi also has an expansive community with a plethora of resources and examples from previous developers, so navigating the system is made much easier.

3.3.6.3 Raspberry Pi Zero WH

The pi zero is a raspberry pi with a much smaller form factor than the other boards in the product line, however it can have its uses in this project. This board can be purchased for a low cost of $5 and can be used to drive the camera in the case that we use a non-raspberry pi computer as the main board for facial recognition. The inclusion of this board allows us to choose more powerful computers for our main device that will be executing machine learning.

3.3.6.4 Nvidia Jetson nano(2GB)

The Jetson Nano is a powerful microcomputer that was created by Nvidia with the intent of being able to run neural networks and machine applications. The Nano contains a 128-core Maxwell GPU @ 921 MHz which is more powerful than the GPU of the model 4. The Jetson with this GPU is able to run many machine learning software like OpenCV, TensorFlow, and Caffe. Nvidia also provides a developer-friendly interface and development environment which is beneficial to get the nano working quickly and makes the Jetson Nano a very viable option.

3.3.6.5 Microcomputer Selection

Table 7 below shows the comparison between all four Microcomputers options and the final selection of the MCU being used.

Micro	Raspberry pi model 3B+	Raspberry pi model 4	Nvidia Jetson Nano	Raspberry pi Zero WH
Operating Voltage	5V	5.1V	4.75 V	3.7V
CPU	1.4-GHz, 4-core Broadcom BCM2837B0 (Cortex-A53)	4-core Broadcom @1.5-GHz	Quad-core ARM® A57 @ 1.43 GHz	1-GHz, 1-core Broadcom BCM2835 (ARM1176JZF-S)
GPU	Broadcom Videocore-IV	500MHz Videocore VI	128-core NVIDIA Maxwell	Broadcom Videocore IV

Memory	1GB LPDDR2 SDRAM	(2-4GB) LPDDR4-3200	2 GB 64-bit LPDDR4 25.6 GB/s	512MB of *RAM*
I/O ports	4 x USB 2.0, HDMI, 3.5mm audio	2x USB 3.0, 2x USB 2.0, 1x Gigabit Ethernet, 2x micro-HDMI	1x USB 3.0 Type A,2x USB 2.0 Type A, USB 2.0 Micro-B, 1x MIPI CSI-2 connector 1x HDMI	1x micro-USB, 1x mini-HDMI
GPIO Pin Count	40	40	40 pin-header + 12 pin + 4 pin fan headers	40
Wireless Comm.	802.11ac, Bluetooth 4.2, Ethernet	802.11ac / Bluetooth 5.0	802.11ac wireless	802.11n / Bluetooth 4.1
Power Consumption	1.9-2.1 Watt	2.85 *Watts*	5-10W	0.7W
MSRP	$35.00	$55.00	$60	$14.00

Table 7: Comparison of Microcomputers

The four microcomputers above offer a variety of different options for our system and what we wish to accomplish with it. Even though the Raspberry Pi Zero WH is the cheapest, it does not provide us with enough features to implement our systems. All the raspberry pi MCU components are also really hard to find at the moment, which is the reason why we decided to go with the NVIDIA Jetson Nano(2GB). NVIDIA provides us with multiple ports, a good amount of memory and a decent CPU to help us run our systems. Also since all team members are currently undergrad seeking full time positions, a lot of hardware jobs use NVIDIA products, so getting exposure to this MCU will be a nice addition to add to our resume to find these full-time positions.

3.3.6.6 MicroSD Card Selection

The Jetson nano requires an SD card to boot the OS just like other microcomputers such as the raspberry pi. Unlike the Raspberry pi however the 2GB jetson nano will write more often to the SD card than the versions with more memory. Because of this Nvidia requires a size of 64GB card and also recommends cards with a high endurance rating to reduce the probability of a card failure due to excessive reading and writing operations.

3.3.8 Software Technologies Selected

Various different pieces of software are going to come together to ultimately form and shape the Indoor BLE security system. As stated, before a big part of this project is the facial recognition system, we plan on implementing on the camera system. The team came to the conclusion earlier on that we wish to use MERN Stack technologies for the web and mobile development portion of our project. MERN stack uses JavaScript as its primary language and the team is very familiar with it. In this section we explore which computer vision and machine learning framework we plan on using and the best approach/database to help train it.

3.3.8.1 Computer Vision Frameworks

There are a ton of different frameworks out there for machine learning and computer vision. Applying these frameworks and figuring which would be best for our needs is crucial because we have various time constraints going forward. Below are the most popular frameworks available for computer vision.

3.3.8.1.1 OpenCV

One of the most comprehensive computer vision/machine learning software libraries out there is provided by a computer program known as OpenCV. While initially having been written in C/C++, OpenCV has branched out to support the Python programming language. One of the benefits to using OpenCV is that their libraries are entirely open source, meaning it is free for anyone to use. The platforms OpenCV provides support for includes Windows, Mac, Linux, and Android. Furthermore, the program provides interfaces for Python, Java, and MATLAB, with OpenCL and CUDA interfaces active in development at the current time. According to their 'About' section on their website, the OpenCV library "has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms." To list a few applications of the algorithms they provide, they can be used for face detection, object identification, movement tracking on cameras, and acquisition of 3D models from objects.

3.3.8.1.2 TensorFlow

Similar to OpenCV, TensorFlow provides a free, open-source software library primarily pertaining to the machine learning and artificial intelligence fields. Platforms officially supported by the software include Windows, Mac, Linux, JavaScript, and Android. The languages capable of using TensorFlow libraries are Python, with support for Java, C, and Go using language bindings. While its libraries can be applied across a wide range of projects, many often utilize the software to aid in the development of neural networks of large-scale.

3.2.8.1.3 DeepFace

DeepFace is a “library for the Python language that implements face recognition alongside facial analysis in a lightweight fashion” [8]. It achieves this through the use of accurate models like OpenFace, DeepID, VGG-Face, Google FaceNet and more in a framework. As stated on their GitHub repository of the library, DeepFace is largely based on TensorFlow and Keras. Their repository also provides useful documentation to install the library on your instance, as well as some demos on how to use each of the implemented features.

3.2.8.1.4 OpenFace

According to the OpenFace documentation, OpenFace is a “Python and Torch implementation of face recognition with deep neural networks” [4], an open-source tool that is built for computer vision and machine learning. OpenFace has been used in the past for prior computer vision projects involving facial recognition, it is highly popular because instead of spending years to develop your facial recognition algorithms we get faster performance and high accuracy due to its Torch framework which allows the software to train the software offline - increasing the accuracy of the overall software but constantly training the neural network.

3.2.8.1.4 Computer Vision Framework Selected

As a team we opted to use OpenCV libraries because some of us had prior experience to working with the collection of packages the library has to offer. Not only does the use of OpenCV stem from the familiarity it has towards some of our group members, but we believe that careers we may find ourselves in the future may view our knowledge of the OpenCV packages and be more inclined to consider us for a work position.

3.2.8.2 YouTube Faces

YouTube Faces DB is a popular tool used to test facial recognition software. The database itself was designed to “study the problem of unconstrained face recognition in video” [3]. Being that our project will require our camera to constantly take videos it is important to train our facial recognition software to work well with video and to be trained to recognize objects over a duration of time. This extensive database will provide training and give us a solid backbone to test our project with.

3.2.8.3 Labeled Faces in the Wild

Labeled Faces in the Wild is a “public benchmark for face verification” [33] systems. Basically, this software is an open database of face photographs designed to train facial recognition systems. This database is known to offer thousands of aligned, deep funneled images for training purposes, giving us a good testing structure to verify that our system is working as expected for a photo if we even decide to do a frame by frame stop.

3.4 Parts Selection Summary

This table below shows all of the selected parts from the previous section. All these components will be used to design and make our project come to life. The table below includes our communication technology, our hardware technology and our software technology we plan on using for this project.

Camera:	Raspberry Pi Camera Module
Servos:	Arducam Pan Tilt Platform
Beacons:	Dialog DA14531 BLE Beacon
Tags:	Dialog DA14531 Asset Tag Card
Microcontroller:	ESP32
Microcomputer:	Nvidia Jetson Nano (2GB)
Communication Technology:	BLE
Database:	MongoDB
Computer Vision Framework:	OpenCV
Face recognition Photo Training Database:	Labeled Faces in the Wild
Face recognition Video Training Database:	YouTube Faces
Web/Mobile Application Stack:	MERN Stack

Table 8: Overview of Selected Parts for Project

To reiterate our selections, our group had various meetings and discussions to go over our parts to ensure that we were all in agreement to why we selected the corresponding parts. For the Camera, we knew that raspberry pi was the optimal choice due to the fact that it has been widely used for other facial recognition projects in the past before, in the ECE department of the University of Central Florida. The camera offers 12MB resolution which is more than good enough for the image quality, they also offer high frames at around 30 fps per 1080p which seems to be standard for most cameras in the $60-$80 price range. Also, we decided to go with this specific camera module due to the fact that we can make the camera its own entity. This mean that instead of having a camera that attaches to our laptop via USB, this raspberry pi will be connected to the microcontroller which will power on the system and using it will deploy our facial recognition algorithm on the computer instead of depending on the computer USB connection to run it. We do have to consider the SD card size however, as those programs get lengthy. The camera module has a lot of libraries that it is compatible with, giving us an extensive option list for which computer vision framework to select. It autofocus feature makes it a great fit for our future software plan, as it will give the software an easier time to run recognition software when able to get a clear picture of the autofocus. In The event that we get stuck learning how to use the camera, we purposely selected this module because of the abundance of work that has been done in the past with cameras.

After the Camera selection, the servo was easy to pick. We wanted a servo platform that was compatible with our camera, so that we know we have good and accurate movement and torque. The Arducam Pan Tilt Platform was the perfect selection, it has the appropriate size for the raspberry pi camera, and it has been used in the past for facial detection projects. So, we knew that this servo and the camera had the capability to work together. The servos and camera platforms have been used for various security applications in the past, so because of the handful of projects that used these platforms in the past we decided to go with the Arducam Pan-tilt platform. Overall, the servos will be responsible for putting the camera in the best position possible to identify any individual that walks into its frame.

The biggest selection and the easiest was the Nvidia Jetson Nano. The Jetson Nano is compatible with both our Camera and our Servos which is why we decided to go with that selection, also working with an Nvidia product is a great plus for our resumes so a big win for that criterion. The Jetson Nano has a decent CPU with a good memory system to make it possible to run these two systems on it. We plan on using a USB connection to extract data from this system at first, but we plan for the future to use UART communication to transmit data from each subsystem. The Nvidia Jetson Nano excels in image classification and object detection projections, and we also have references to multiple computer vision projects that have used both Nvidia Jetson Nano and Raspberry Pi HD camera.

For our Bluetooth beacon our team felt comfortable with the Dialog DA14531 BLE Beacon due to its association with indoor tracking. Although we have not found many projects that implement this beacon, our team is confident enough to take on the task. Its long broadcast rate made it an easy choice, as well with its relatively low price. A big plus was its support for iBeacon and Eddystone protocol, which allows us to switch between either of these if one of them does not work out. Feasy is also a great provider because along with the beacon we get sent software and hardware packages that can be customized to our use, so the implementation should not be too difficult, we just want to ensure that the signals are strong enough to emit to both the tag and the microcontroller.

The Microcontroller we selected was the ESP32. The ESP32 is something we are excited to use and learn. This MCU is integrated with Wi-Fi and Bluetooth connectivity which gives us the option to transmit data via Wi-Fi if UART does not go according to plan. The ESP32 Bluetooth integration means that it will be able to act as a tag, so our primary focus will be getting this tag to receive data from the beacon to transmit over to our IOT service. This microcontroller also has super low power consumption making it long lasting and fit to act as our Bluetooth tag for future implementation. This chip from our understanding has not been used in any indoor tracking application, however we are confident that with its Bluetooth module that we can easily incorporate it into our project, we do expect to run into issues and are prepared with troubleshooting steps when said issues arise.

For our testing purposes our team has also planned to acquire the Dialog DA14531 Asset Tag Card. The Bluetooth tag is mostly for cover and testing. We want to test if the beacon is actually able to emit signals and that tag is capable of receiving these signals. This tag goes hand in hand with the beacon we selected so we know that they both work using similar protocols, the tag also comes with NFC features which is great for future security implementation to add on if time allows it. The tag and corresponding beacons have been used before for indoor and outdoor location applications which was a huge plus for us since we want to implement, relatively the same thing. Feasy also comes with preloaded software available to test and configure any settings which is nice for future edit and changes. Ultimately this tag was chosen to test our beacon and if our microcontroller implementation does not go according to plan then we have aback to use that we know works and has functioning software on the side that we can alter for our usage.

As stated, before we also choose to go with the MERN stack for our mobile and web application. The MERN stack consists of the following technologies: MongoDB for database, Node.js for server and backend, Express for backend, and React for Frontend. We decided to go with MERN Stack due to an easily scalable solution, so we can start off with something simple but if more complexity or more information is needed in general out of the site then we can easily do so with the MERN stack. Also, the MERN deployment is simple and free with the help of Heroku. We do want to know that everyone in the group is familiar with this software stack and how to implement CRUD APIs such as create, delete, update and search. Previous projects done in the past have used MERN and LAMP stacks, so these two are among the most popular. We could have gone either way with a stack, but we believe that going with the MERN application is the easiest and also the most up to date in terms of technologies.

Finally, we choose our computer vision framework, OpenCV which is a very popular library that is often used for image processing and computer vision tasks. OpenCV supports multiple languages such as python and C++ which gives us the flexibility between the two languages. As of now, the team has decided to go with Python for the main language of programming. Again, OpenCV is an extensive and popular library that has been used multiple times in the past as well for other projects, so again we are confident that we can follow in the right steps and come up with a solution for our facial recognition system. Also given that there are so many other frameworks out that also have a lot of projects done within their library, if we ever run into an issue, we have the availability to switch frameworks and try something new. We have discussed that our second option will be OpenFace, but we hope everything goes according to plan.

For our training modules we choose YouTube Faces database to train our facial recognition module with video-based image capture so that it learns to recognize faces in a situation where we must upload a video to our software to run an analysis. We also decided to use Labeled faces in the Wild as another training database, which uses images instead of videos for its training. We hope to have our software recognition working with both video and photos when the recording sessions are established for our smart security system. Having to train our module using both different training databases will come in handy whenever our system either records a video of individuals when they come in the frame or allows us to stop the images frame by frame to acquire a single photo in which we would run our analysis.

Ultimately these are all the components and parts we choose for this project. This is a scenario that our group possibly did not account for everything and perhaps forgot some minor components to go hand in hand with other components we selected, in that case we give us a little more in the budget section to cover that scenario if anything were to happen. Also, the part components do not include any hardware testing materials such as breadboard or multimeters as those were provided to our group by the discovery kit available to all senior design students by the College of Engineering and Computer Science at the University of Central Florida. The bulk of this project will be built and stored at the Senior Design Lab Room 456 of the Engineering 1 building at UCF. Our team decided to perform our building and testing in this specific workspace due to its location and it's a good meetup place from Home for each team member and second being the utilization of the schools’ facilities and their testing equipment specifically for our own usage. One top of the senior design lab our team also has access to the Texas Instrument Innovation Lab which has similar equipment for testing and more tools for combining components. Overall, our team is content with the components selected and have been waiting for the components to come in so we can be testing them to ensure all of them come in working shape, we did this earlier than us anticipate date due to the fact that we do not want to be in a situation where we fall behind and have to rush to get everything done. Once again, the senior design lab is the primary space where all our components will be held and tested as it is a common ground for all of the team members.

3.4.1 Final Parts Selection summary

In our final prototype we made a few changes to the parts discussed before. When testing the computer vision component of our project we found that the Jetson Nano 2GB was not suitable for real-time data transmission. This issue arose from the small amount of RAM that the Nano had. To remedy this, we would have purchased the standard 4GB Nano if it were in stock. Due to not using the Nano a number of other components had to be reconsidered. For example, rather than using the raspberry pi camera we instead used the Logitech C920s since we decided to train the computer vision code on a desktop computer. Due to not using the raspberry camera we also omitted the servers from our project. To conclude, the failure of the Jetson Nano caused us to omit a number of components from our system, however we were still able to achieve the goals of our project.

4 Related Standards & Realistic Design Constraints

Standards specify characteristics and technical details that must be met while implementing certain technologies into your design. Each piece of technology being used here has a standard or technical detail that describes how the product must be used in terms of limitations and operation of the design. This chapter will cover the standards and design constraints that our team will be focusing on while researching and designing our project.

4.1 Standards

IEEE 2600-2008

IEEE 2600-2008 is a standard that defines security requirements for all aspects of security including authentication, authorization, physical security and information security. This standard instructs users/manufacturers to set appropriate security capabilities and to provide instructions on appropriate ways to use these capabilities.

IEEE 208-1995

IEEE 208-1995 is a standard that defines methods of video technique and measuring of resolution of camera systems. Its primary application is for users/manufacturers to quantify the max limit where fine detail contained in the original image is no longer reproduced by the camera system.

P2884

P2884 is a standard that defines requirements and methods for facial recognition testing systems for end user devices. Some examples are algorithm testing, security requirements and testing methods. Key metrics are given to evaluate performances such as false acceptance rate, and false rejection rates. This standard evaluates the performance of our biometric facial recognition system and holds us up to standard with the indexes provided.

P2716

P2716 is a standard that provides manufacturers and users of printed circuit board level shielding with appropriate methods for the characterization of the shielding effectiveness of the board level. This standard goes over various techniques of how to mount these shields and describe how to test your selection by providing various methods depending on the intended application.

IEC 61188-6-1:2021

IEC 61188-6-1:2021 is a standard that specifies the requirement for soldering surfaces on the circuit boards. This general principle is applied by setting requirements for land patterns on circuit boards.

IEC TR 61191-7:2020

IEC TR 61191-7:2020 is a standard based on the technical cleanliness of components and printed board assemblies. This standard provides information on how technical cleanliness can be accessed within the electronics assembly industry. Following these general guidelines to analyze the associated risk of electronic assemblies in the electronics industry.

IPC-2615

IPC-2615 is a standard based on printed board dimensioning and tolerancing of electronic packaging. These standard assists manufacturers and customers achieve the best board design possible by following the fundamental dimensioning and tolerance rules based on IPC-D-300G.

IEC 60068-2-64

IEC60068-2-64 is a standard that is used to test the integrity of embedded and integrated circuits when subjected to vibrations. The purpose of this standard is to ensure that manufactured boards are able to withstand varying levels of movement/vibration without the mechanical integrity of the device being reduced.

EIA-364

EIA-364 is a standard that is used to test the integrity of electrical connectors and sockets of devices in a controlled environment. This standard's purpose is to ensure that there is a minimal acceptable level of integrity when it comes to inserting and removing connections from pins and sockets. For the Nano this ensures that pins such as the GPIO pins and the external peripheral connectors are strong enough to be used by users handling the device.

IEEE 802.15.1-2002

IEEE 802.15.1-2002 is a standard based on telecommunication and information exchange between systems specifically created for Bluetooth wireless technology. These standard lays specification for wireless personal area networks and description languages model for integrated lifecycle. This standard helps reduce the likelihood of defects and improves overall project management which will be useful for our agile approach.

ISO/IEC 30137-1:2019

ISO/IEC 30137-1 is a standard that monitors the use of biometric in video surveillance systems. The biometric most popular for video surveillance happens to be face recognition which is what our group is implementing. This standard provides guidance on modalities such as gait recognition and includes metrics for defining performance to see where our system stands. ISO/IEC 30137 also has recommendations on data format for facial images and establishes principles for supporting user interfaces to promote efficient and effective operation.

ISO/IEC 19794-5:2005

ISO/IEC 19794-5 is a standard that focuses on biometric data interchange formats, specifically face image data. This standard sets specification on the format requirement for images of face to be used in the context of human verification. It shows us the best practice methods to layout guidance on photo capture of a system, using ISO standard JPEG or JPEG2000.

ISO/IEC 18305:2016

ISO.IEC 18305:2016 is a standard based on real time locating systems. This standard provides test and evaluation of localization and tracking systems, and helps users identify appropriate performance metrics. It sets fundamental guidance and evaluation scenarios based on indoor environments.

IEC 62676-5:2018

IEC 62676 is a standard for video surveillance systems for use in security applications and focuses on data specification and image quality performance for camera devices. This standard provides recommendations and requirements for representation and measuring methods for surveillance camera equipment. It helps users and manufacturers define requirements for description of the specified item being recorded and how to go about the measurement methods.

4.2 Realistic Design Constraints

Due to budget and a timeline of one year there will be areas of the project that will be restricted. This project is not funded by a sponsor and the personnel completing the project have little experience in designing and implementing a project consisting of multiple components such as a PCB, microcontroller, and microcomputer. The biggest constraint impacting the design of the project is the inexperience in PCB design across each member of the team. Course material from Junior Design is a great reference for finding parts for a board, simulating the functionality of the board, and utilizing the data sheets. Another added layer that adds constraints to our realistic design is the chip shortage. There may be parts required for the PCB that may work perfectly with one another but are out of stock everywhere. It could very well be possible that our design suffers in the implementation as a result of the shortage as well. One missing part could misalign the compatibility of important components of the circuit. In ongoing research for beacons to implement in the security system it is very evident that suppliers generally do not sell beacons for the average consumer needing a small number of beacons. In fact, companies found in our research require quote requests to get pricing on beacons.

4.2.1 Economic and Time Constraints

To make the execution of the project more efficient our team decided to establish rough deadlines and a timeline for certain parts of our project. In doing this we have some of the major components of the project finished prior to the beginning of the spring semester. The idea is that this will help alleviate the workload in the later stages of the project, which will give us some adequate time to account for any slowdowns in development that might occur. One self-imposed time constraint is that we planned to have the web application MERN stack setup so that when we implement the location mapping all of the necessary infrastructure is already in place. We also planned to have the facial recognition code training done by December so that when we decide to implement the pan-tilt-zoom digitally it can be done quickly as the facial is already set up.

As for constraints that are out of our control, we have need ample time to establish facial recognition and train the network which could take up to 2 weeks to perfect. We also have to keep in mind that due to the pandemic the world is undergoing a severe shortage of microchips, and this has an effect on our group procuring materials. For example, Raspberry pi has experienced massive shortages and is sold out on almost every online marketplace, so we have had to pivot to use the Nvidia jetson nano due to scarcity.

Another economic constraint we face is that our project is self-funded and not backed by a sponsor. Our estimated cost is approx. $400 so we are aiming to keep the cost low where we can. For example, it is possible to buy commercial BLE position tracking hardware, however these systems can cost thousands of dollars in setup and upkeep. These high prices were not feasible at all for our implementation and scale, so we have chosen to create our own BLE beacons using a PCB BLE module. Taking this approach to the project helps to save money that can be better spent on other crucial hardware systems.

Our team also had to consider the tradeoff of costs vs quality for the camera that we decided to use for facial recognition on the Nvidia Jetson Nano. We viewed and compared the cameras based on the best possible image quality for a relatively low price. Due to the 2020-2021 Coronavirus pandemic the team has been unable to meet, and this does have an effect on our timeline as we aren't able to easily meet in-person to work on the project. This also can conflict with our class and work schedules, which can make it difficult to set up meeting times to work on the project as a group- person. Below in figure 8 and 9 you will see the expected schedule and milestones for Senior Design 1 and Senior Design 2 that we as a team are expected to complete:

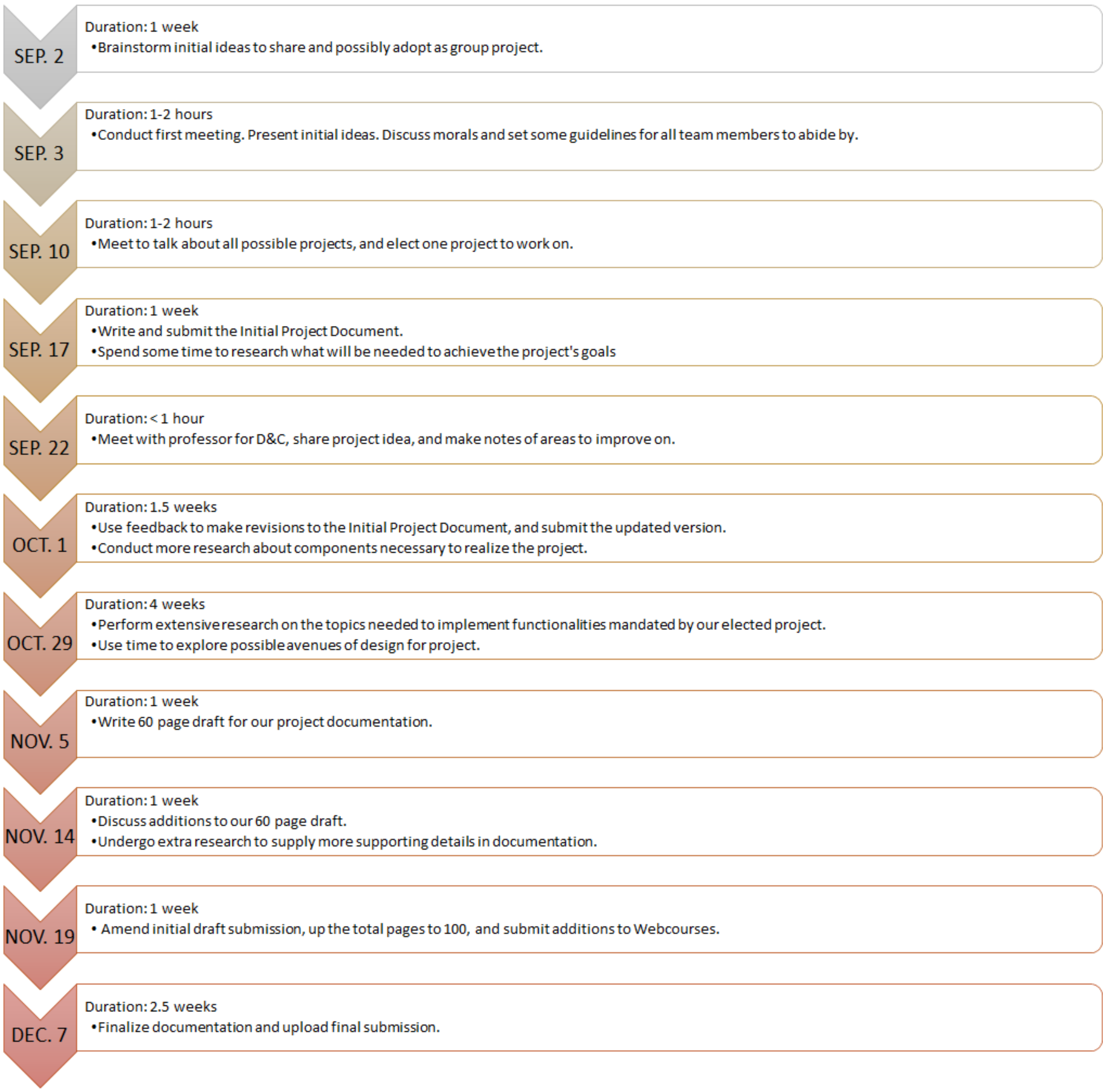


Figure 8: SD1 Milestones

Figure 9: SD2 Milestones

This table below shows our expected costs for completing this project, as mentioned before this project is self-funded so our parts were chosen according to what we found reasonable as a group:

Item	Est. Price
Microcontroller for camera x 1	$6-$20
Microcontrollers for receivers 2-4	$20-$40
Central Microcontroller (Hub)	$20-$30
Custom PCB boards for tags	$50-$110
Wiring pins for microcontroller	$10-$20
Camera(s) used for facial rec.	$15-$22.50
Mount for cameras and Beacons	$20-$30
2x Servos for mounting	$8-$20
Batteries w/ accessories	<$30
Circuit elements (Breadboard, resistors, LEDs, etc.)	self-sourced
Total(estimate)	$170-$380

Table 9: Estimated Project Costs

4.2.2 Environmental, Social, and Political Constraints

Due to the nature of the project, the implementation of the design is limited to indoor tracking only meaning that there will be no outdoor tracking available. Another limitation on the design will have to do with protecting individual privacy. Because of this, the indoor positioning system will not include private office spaces. Bluetooth signals have their own constraints so our field of implementation will also be limited by these constraints. The signal distance limited by BLE bands could be within the range of 10 meters (33 feet). Wi-Fi will also be included. With that, the signal will be limited by both the interference and the power output to the Bluetooth device. Ultrawide bands do not have the same environmental constraints as low energy bands; however, the power consumption of the ultra-wide band is a constraint because we would like this to remain low. The less battery required to power our signal the better, but in our implementation ultra-wide band seems like the better option although it consumes more, and the same size battery could potentially drain substantially faster.

Social constraints are a bit more subjective than objective. People on-site may feel an objection to being watched like any other person, but there will be guidelines in place to ensure the safety of anyone entering the building and being watched by the cameras. An issue that we are avoiding by planning to set cameras up in the corners of the room as opposed to having an individual hold a camera up to each individual as they enter a secure area is giving personnel room to breathe as they enter a room. Another constraint that could potentially affect the implementation is the willingness of personnel to voluntarily submit to tracking within a facility. Each participant may be required to sign a waiver to have their real time position tracked within a facility they have access to. There could potentially be a breach in privacy by temporarily storing pictures of individuals that do not have an entry in our database. Though the access to this information is private, the nature of its legality is still in question and further research needs to be completed before this potential constraint may be removed from consideration.

Political constraints impacting the implementation of the project is also a social one, but politics weigh more in the outcome of this constraint. Due to the slowdown in production as a direct result of COVID-19, many manufacturers are experiencing chip shortages. Governments forced facilities to temporarily shut down to flatten the curve. In this short time, chip manufacturers were forced to close their doors as the demand grew. To date, the chip shortage is still having a negative impact on sales relating to computational devices. Our project is negatively impacted by the shut down because it limits the options for us in regard to the microcontrollers, microprocessors, and the parts we need to order as part of our PCB design.

4.2.3 Ethical, Health, and Safety Constraints

4.2.3.1 Ethical Constraints

We have set out to develop a security system that ensures there are no intruders in the environment the system was installed in. With that, the ethicality of the systems comes into question. On one hand, using cameras and keeping track of users' location definitely puts the privacy of people involved in jeopardy when they're in the area. There would be a considerable amount of trust put on the thing that is responsible for managing the entire system, whether it was just a machine, or even a human monitoring the system. One application of our system would involve people working in the location where the system is installed, and their whereabouts would need to be monitored to ensure they are not accessing locations they should not be. At the time of writing, we believe the strongest approach involves having each employee keep a uniquely identifying card on them throughout their work time. In this way, employees would be less encouraged to try and enter areas they are restricted from. Another point of concern may be when it comes to the cameras being utilized. Again, to some it may be uncomfortable working under the surveillance of cameras, especially if other people could potentially be accessing the film. In our design, cameras are strictly being utilized to cover the faults that could occur if two cases occur. The first case comes when outsiders who are not employees at the location attempt to enter, and the camera will observe this when it fails to recognize the face it is observing of the intruder. The other case occurs when an active employee is not carrying their card on them and finds an alternative way to enter areas they are restricted from; the camera would identify a face that it was not expected to enter at that moment. In both cases, the camera would raise a flag or warning to some administrative person in charge and only then would a real person be looking at the footage. The idea here is trying to minimize the amount of time a real human would need to spend monitoring the system, and that will involve building a decently reliable machine that is capable of identifying faces.

We have considered that the system may not be something everyone is willing to subject themselves to, but we believe there are some workplaces where the system would be accepted and necessary. The most important thing we stress for companies that would decide to implement this system in their workplace is that they are fully transparent to each and every employee working there. People already working there would need to be informed before the system gets integrated into their work environment, so that they can make decisions about whether or not they want to continue working at that company. Likewise, any new prospective employee would need to be fully disclosed about the tracking system, ideally in a discussion, such as an interview or phone call. We believe that so long as everyone working there is comfortable with the system in place, it would be a completely valid system to have, in order to ensure security at the workplace is robust. In that sense, some of the ethicality would come from how each company approached informing their employees, as well as their efforts in communicating with future members.

4.2.3.2 Health and Safety Constraints

With regard to the health constraints in our project idea, nothing will truly be impacting the overall health of people enveloped within the system. Our system does not require any invasive treatment to operate, and the only thing required to keep track of people in our system is the card each person is provided by the administrators. The benefit with this implementation allows any new person to easily become connected to the system. It should be just as easy to remove a person from the system, should they decide to not involve themselves in the place the system is installed at anymore.

As far as safety constraints, there are a plethora of things to consider, some of which have been mentioned in the section above. Information safety is something we want to make sure every member involved in the system has, as they would have allowed us to store the information, with hopes that it is only being used to keep the location secure. Possible concerns with the information leaking could arise from anyone involved, and we understand this concern as we are responsible for ensuring the information shared is kept confidential. The steps we are taking to make sure the information is secure involves implementing and requiring a two-factor authentication mechanism in our application, that way users have to verify the user attempting to login to the web application is not an intruder trying to access private information.

4.2.4 Manufacturability and Sustainability Constraints

In terms of material and parts, the team is not planning on manufacturing our own components for this project due to the time it would take to complete this process. For the hardware components, we plan on gathering modern and reliable parts to use for our projects. We plan on buying the hardware components from reliable vendors. In terms of software, we plan on using the most popular and most documented options for our facial recognition software and our database. The licenses for our software are open source, which allows for redistribution, and modification of these source codes.

Durability is also an important aspect, we want to create a system that can be used for a good amount of time, and reliable in a sense that systems work as expected most of the time, given that it has not taken any structural damage. Battery life plays a factor in our Bluetooth beacon and Bluetooth tag system, which is planned on being purchased by reliable and well documented manufacturers. We understand that tags should operate for a reasonable amount of time without having to replace the battery. Therefore, as we review possible avenues for acquiring these components, our group will carefully consider this specification on each tag we provide for users of our system.

It is important that we keep hardware such as beacons, camera platforms and tags in safe locations at all times to prevent the integrity of our system from being compromised. To realistically account for this concern, we suggest storing these crucial components of the system in a well-restricted location where the system is installed. For optimal security, however, we advise the location of any spare beacons, cameras, or tags at an external location.

Advising how to use and handle these hardware components will reduce risk of negatively affecting the system. We plan on providing clear suggestions on how to properly place these cameras throughout the relevant areas.

5 PROJECT HARDWARE AND SOFTWARE DESIGN

This section will cover the in-depth detail of the project outlines, describing how each component will be designed and constructed. This section will be split into two parts, the hardware design and the software design. In the hardware design section, we have covered the indoor system with the Bluetooth beacons along with power and antenna components and their implementation. The software design section will cover our plan for the facial recognition system, and a deep dive into the user interface we plan on creating with the web and mobile application.

5.1 Hardware Design

This section will cover the hardware design for the indoor BLE security system. You will see throughout the section various diagrams and schematic for each system and component, the team will go into detail about how each part will work and interconnect with the other working parts. You will read about why each system is crucial for our system and how it connects with the other components listed, providing clarity about project prototypes, assembly, and usage.

The diagrams you will see were made by the team via diagrams.net. As mentioned before various schematics will be seen - these representations were constructed in Eagle CAD, our primary choice for circuit modeling. We have covered the design and usage for our PCB board and how it was constructed. Altogether, the following subsystems are the primary hardware design for our security platform.

5.1.1 Camera Module Overview

One of the two major hardware components in this project will be the camera mount that is used in our facial recognition and user identification. The figure below will show the basic connections between the hardware within this camera module. Not included is the schematic showing the camera in a 3D environment.

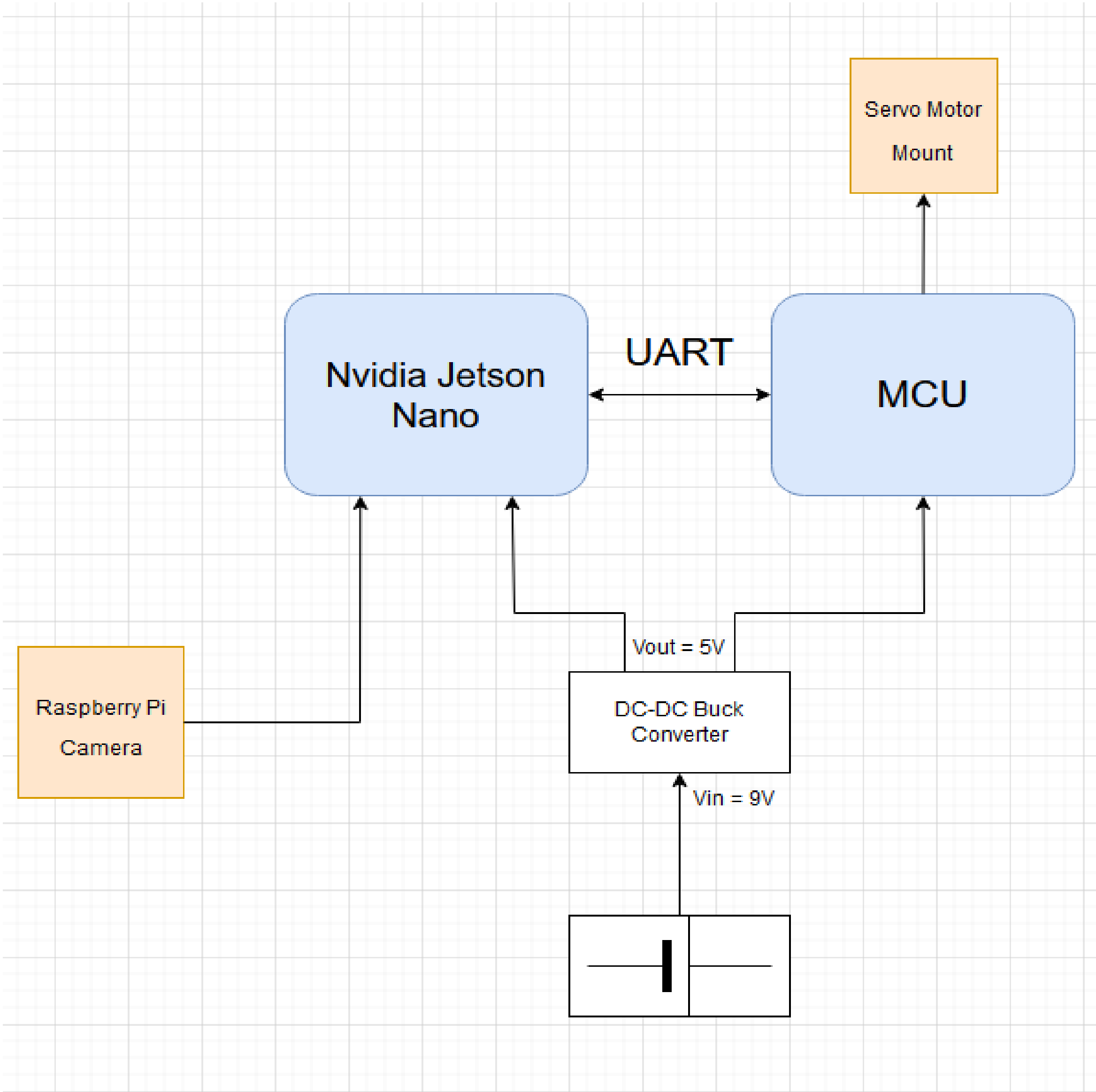


Figure 10: Camera Module Block Diagram

This diagram demonstrates the way that the hardware within the camera mount will be connected together. The Jetson Nano will be the main driver for facial recognition and will have the raspberry pi camera transmitting a live video feed to it. With this information it will be constantly communicating with the MCU in order to power the servos to implement our pan--tilt- zoom tracking algorithm. The MCU's main function will be to provide PWM signals to the servo mount in order to move it according to the findings of the camera. The power for both of these components will be provided through a 9V to 5V buck-down converter.

Ultimately, due to the difficulty of getting our face recognition to work on the Jetson Nano (2GB) we had to change this approach and scratch this from our implementation.

However, we believe that if we were to purchase the 4GB version, that we should still be able to follow this block diagram and implement our original design as intended.

5.1.2 Beacons and Tag Functionality (Trilateration Implementation)

The beacon and tag system are the foundation of our indoor tracking system. Our group decided to order the tag and beacon discussed in the part selection above for testing purposes to study more about the connection and its Received Signal Strength Indicator (RSSI) value which is what will be used to start our trilateration calculation. Also, previously we discussed previous IPS methods, such as Angle of arrival, triangulation and Trilateration. Our team decided to move forward with Trilateration due to the abundance of resources, studies and practice available for use.

To implement trilateration, we need three beacons and a system of tags. Our three beacons will be responsible for emitting Bluetooth signals to detect these tags in our security system. These beacons will be composed of our ESP32 microcontroller selection which our group plans on using as our main MCU for our PCB design. These Beacons, once they receive a signal from the tags, our program will take and store the given RSSI data from the tag, its mac address and store it in a payload to send over MQTT.

Trilateration uses distance estimation against RSSI data to indicate where our tag is in reference to the three stationary beacons. The three stationary beacons all hold an (x, y) coordinate and based on signal strength from the tag in comparison to the coordinates, estimated coordinates for the tag are calculated. Trilateration requires multiple steps before the algorithm reaches a result that accurately represents indoor position within 1 meter. Firstly, the RSSI value from the incoming tag is received from beacon. The RSSI is calculated using ESP32's built in getRSSI () function. Once the RSSI value from the tag to its respective beacon is transmitted, the data is transferred over to the software application using MQTT where the data will be manipulated.

Next, is to calculate the distance from the respective tag to beacon using that incoming RSSI value (1).

$$d = 10^{((TxPower\ -\ RSSI)\ /\ (10\ *\ n))} \qquad (1)$$

The distance formula consists of three parameters that have to be accounted for. The first parameter TxPower, is known as Measured Powered, in other words the 1-meter RSSI [6] which for the ESP32 module is -60 at 1 meter. Next is the value of the incoming RSSI from the detected device. In this case, the tag is being detected by beacons and emitting respective RSSIs readings to its appropriate beacon. Lastly, is the 'N' parameter. 'N' is a constant that we choose to set based on the environmental factors. This variable usually ranges from 2 to 4, for our project we ran multiple tests to determine the optimal value of 'N'. Once all the parameters are filled in by the incoming data, we can now calculate the distance 'R_i' from the respective beacons to tag. Each beacon should theoretically receive a different RSSI value unless the tag is in the center of the coordinate grid. Our stationary beacons will each have their own (x, y) coordinates accompanied with a relative distance, where distance will be calculated by the incoming RSSI, and this will form the matrix in equation (2):

$$\text{Tag Position} = \begin{bmatrix} [x1, y1, R1] \\ [x2, y2, R2] \\ [x3, y3, R3] \end{bmatrix} \qquad (2)$$

From here, the trilateration implementation uses a point estimation equation based on Pythagoras Theorem [42], to obtain a resulting coordinate pair (3) to approximate the location of the tag that the system is currently detecting.

$$Tag\ X = \frac{\begin{vmatrix} R1^2 - R2^2 - (X1^2 - X2^2) - (Y1^2 - Y2^2) & 2(Y2 - Y1) \\ R1^2 - R3^2 - (X1^2 - X3^2) - (Y1^2 - Y3^2) & 2(Y3 - Y1) \end{vmatrix}}{\begin{vmatrix} 2(X2 - X1) & 2(Y2 - Y1) \\ 2(X3 - X1) & 2(Y3 - Y1) \end{vmatrix}}$$

$$(3)$$

$$Tag\ Y = \frac{\begin{vmatrix} 2(Y2 - Y1) & R1^2 - R2^2 - (X1^2 - X2^2) - (Y1^2 - Y2^2) \\ 2(Y3 - Y1) & R1^2 - R3^2 - (X1^2 - X3^2) - (Y1^2 - Y3^2) \end{vmatrix}}{\begin{vmatrix} 2(X2 - X1) & 2(Y2 - Y1) \\ 2(X3 - X1) & 2(Y3 - Y1) \end{vmatrix}}$$

Theoretically, the trilateration algorithm gives us an approximation within 1 meter of the actual position. In practice, each beacon reaches within .14 meters of the actual position at our tested distance. Achieving accuracy while remaining computationally efficient (low cost) was important and trilateration made this possible. On average, the response time between movement of the tag to a new location and publishing the change on the software application stays around 1 - 2 seconds due to latency.

5.1.2.1 Beacon Power Supply

The Beacons, we plan on having two power supplies. We want a stationary connection to a USB-C power cord, and also allow the option for easily movable beacon with the implementation of a LiPo battery component.

5.1.2.2 Beacon Antenna Design

For our Beacon Antenna our group opted to use the built-in antenna in the ESP32 WROOM module.

5.1.2.3 Beacon Signal broadcast rate

To preserve the battery, the signal strength will be strong enough to extend omnidirectionally with a frequency band of 2.4GHz.

5.1.3 Camera and Servo Framing

The camera and servo framing are one of the essential components of the system. This bundle of hardware is independent from the beacon and tag system and will be responsible for the camera aspect of the smart security system. The servo we purchased is compatible to hold the raspberry pi camera we selected, as it was built specifically for that camera module. The servo is programmable with the NVidia jetson microcomputer as well, so all the software we plan on implementing will be pushed from there. Below you can see Figure 11 which shows how we plan on hosting the camera on top of the tilt pan platforms. Using 5mm-8 screws we were able to mount the camera on top of the platform.

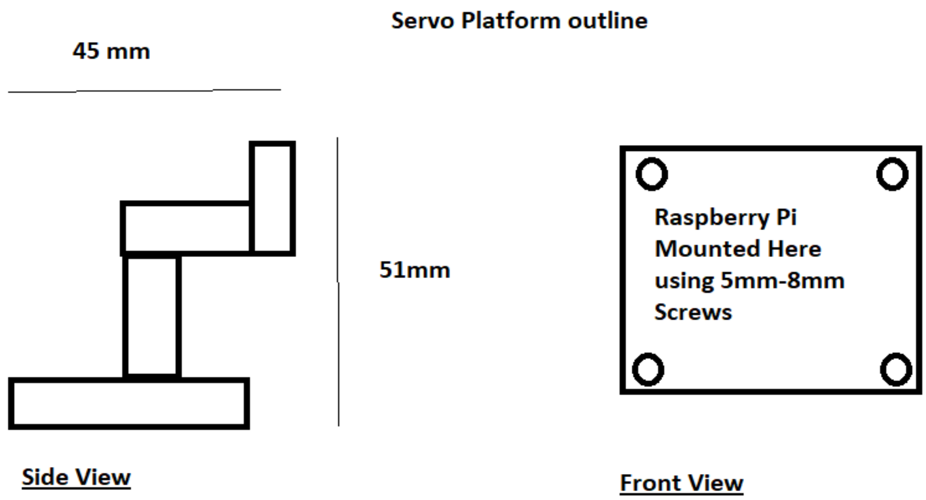


Figure 11: Servo Platform Outline

We plan on using a PTZ control board that comes with the platform to connect it with the Nvidia Jetson. The camera comes with a connection ribbon known as Ribbon Cable Contact, using this ribbon we find the CSI port of the Nvidia Jetson and connect it to the board. Below you will see a diagram of the Jetson nano, we highlight the CSI connection input we are using for the camera module.

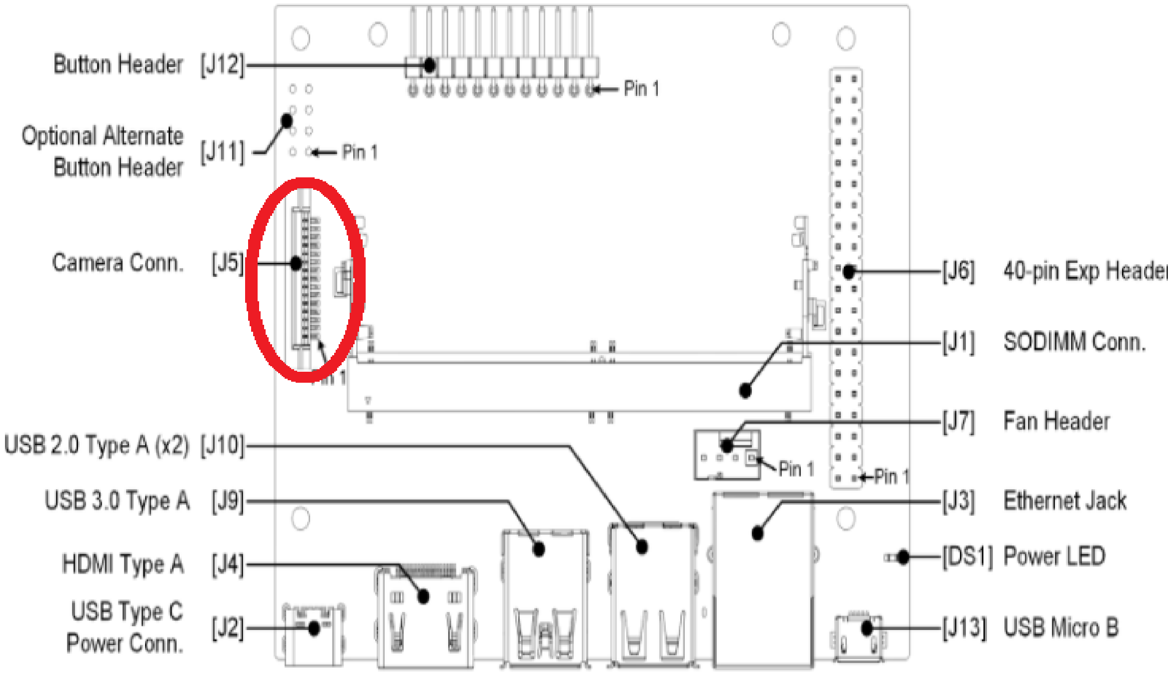


Figure 12: Jetson Nano Diagram

After establishing the connection, we plan on wiring it with the control board, which uses the I2C interface and an operating voltage of 3.3V/5V. By connecting the Control board to the servo, we can also connect it to the microcomputer, in this case the Jetson Nano. Once connected, the Nvidia Jetson can be programmed to control the servo motors and its functionalities. Below in Figure 13 is a sample diagram of how the wiring and connection with the jetson, control board, camera ribbon and servo motors will work.

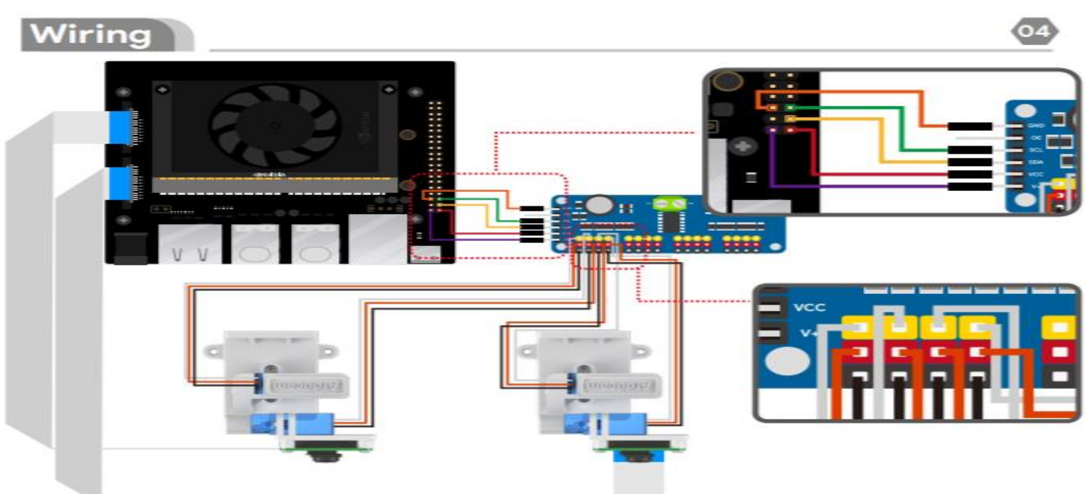


Figure 13: Arducam Nvidia and Servo Diagram [5]

Once connection is established, servos will be responsible for moving the camera left and right and tilting it forward when required. We have program the jetson to control these servo motors and their functions. The camera will not have the facial recognition software directly installed into the module we are discussing about here, since the framework for the facial recognition system is too big, we have instead been using one of the USB ports to connect the jetson with our laptop, providing the system with power. When we have the camera fully ready to operate, we can run our facial detection software in real time and display the results live from our computer screen. In the future when this product is ready to be implemented real time video will then be transmitted over via the IoT network and cloud and uploaded onto our laptop, where we can run our facial detection software and analyze the results. As stated before, our servos will be programmed to stop the camera once an object in the frame has been detected. Ultimately, this system will be mainly responsible for a couple of things, mounting the camera and making sure it can move and stop when an object is detected and for the camera to work with the jetson for the facial detection operation.

5.1.4 MCU PCB Schematic

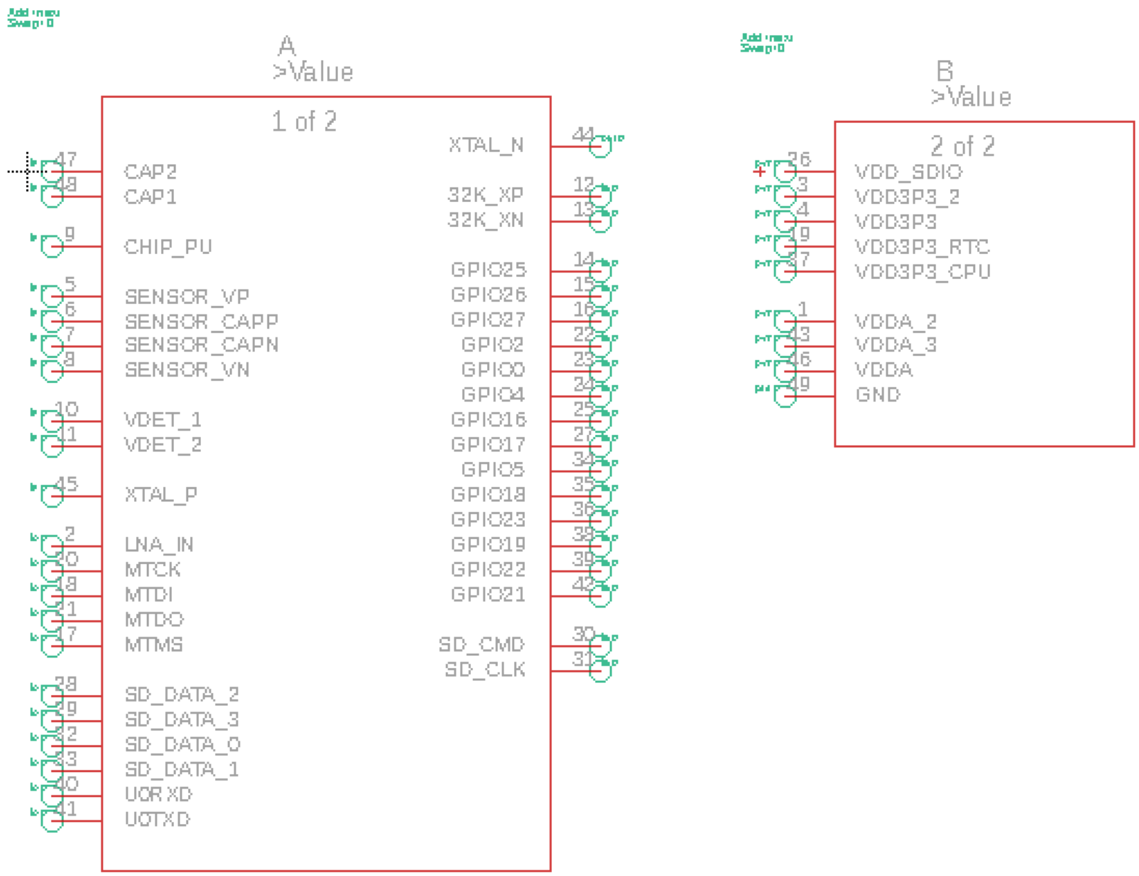


Figure 14: MCU Schematic

MCU Pin Descriptions

These descriptions will reflect the pins that are necessary to utilize as part of the IPS. The sensors along with the power are the pins that will be used in this prototype.

Sensor_VP

Input only GPIO that reads incoming data based on pin configuration

Sensor_CAPP

Input/Output GPIO that writes and sends data based on pin configuration

Sensor_CAPN

Input/Output GPIO that writes and sends data based on pin configuration

Sensor_VN
Input only GPIO that reads incoming data based on pin configuration

GND
Unused pins that impact the functionality of the circuit will be connected here. This will also ground the circuit where necessary to sustain the flow of current and safely monitor voltage

5.1.5 Microcomputer Schematic

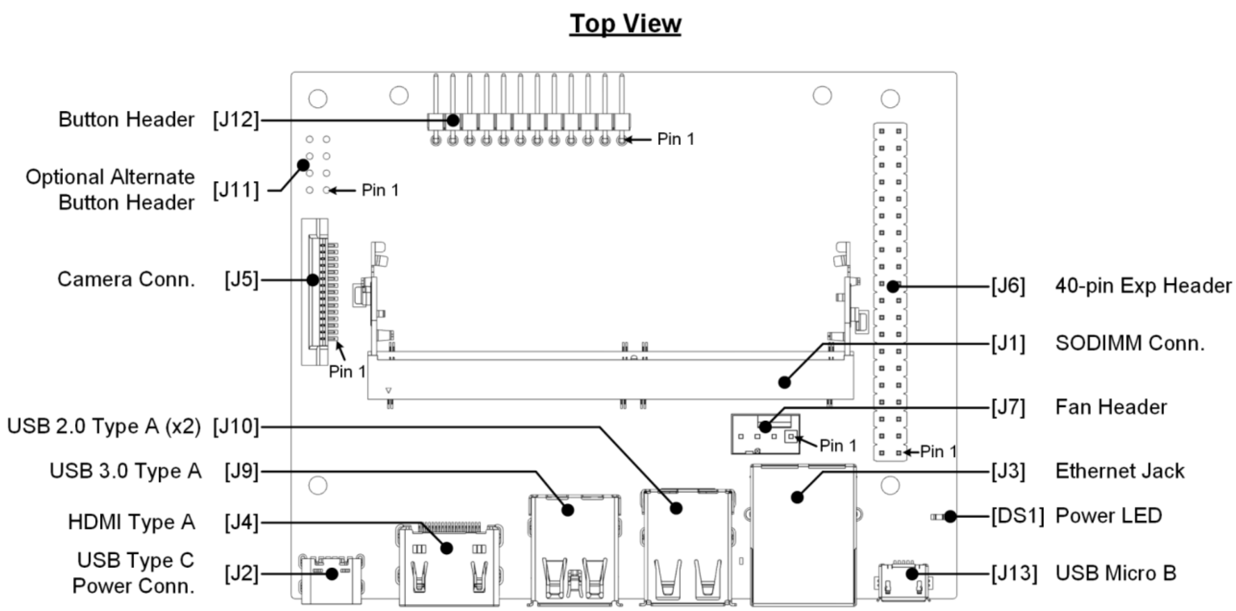


Figure 15: Nvidia Jetson Nano Top-Down view

5.1.6 PCB Schematic

This is a filler schematic for the PCB which will host a socket which will provide power to the Jetson Nano and Camera Module (attached to the microcomputer).

PCB startup

A key component to both our microcontroller and Jetson Nano is the system that powers them. For this we have decided to create a printed circuit board that will be responsible for providing power to both the MCU and Nano boards. The power supply for the Jetson Nano is required to provide 5V and 3A. And for the MCU the power requirements are 2.2V to 3.6V at 600mA.The PCBs will be mounted on the camera mount where the MCU, Servos, Camera, and Nano will be located. And they will fit within the camera unit.

Powering the Nano

As stated, before the Jetson Nano will require 5V and 3 A to provide adequate power so we have decided to create a buck converter that will allow us to deliver the appropriate DC voltage and amps to the devices. The input voltage will be delivered by a rechargeable battery pack to save money on having to purchase batteries over and over. The purchased DC battery pack will not provide the exact 5 volts that we need so we have reduced the input voltage to match that of the devices with the buck converter. A buck converter is a DC-DC converter that allows an input voltage to be efficiently reduced down to a lower voltage level. We have been using a simple converter that will convert a standard 9 Volt battery into the 5 Volts needed.

5.1.6.1 PCB Schematic Design

LM-2576 Step-Down Voltage Regulator

This voltage regulator will be responsible for actually converting our higher input voltage down to the 5V we require. This integrated circuit only requires a minimal number of external parts to complete the full buck converter. This regulator allows us to drive the required 3 amps as well and includes safety measures to stop any failures from harming connected devices.

Circuit Elements:
1x330Ω Resistor

1x1000μF Capacitor
1x100μF Capacitor
1x220μF Capacitor
1x 10KΩ Potentiometer
1x100μH Inductor
1N5822 Schottky Diode

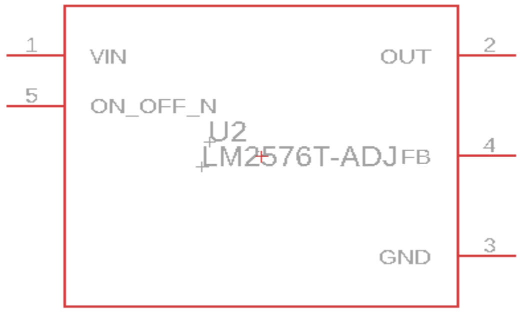


Figure 16: LM-2576 IC diagram

The schematic above is the lm-2576 block diagram and the pins work as follows:

PIN 1, Vin:

The function of this pin is to take the positive input voltage to be converted and send it

PIN 2, Vout:

When the integrated circuit completes the step down the voltage is sent out of this pin

PIN 3, GND:

Standard ground pin of the integrated circuit

PIN 4, Feedback(F/B):

The feedback pin is used in looping voltage systems and will compensate for voltage given through feedback loops

PIN 5, On/Off:

Establishes whether there is a connection or no connection to the step-down converter based on the mode it is placed in.

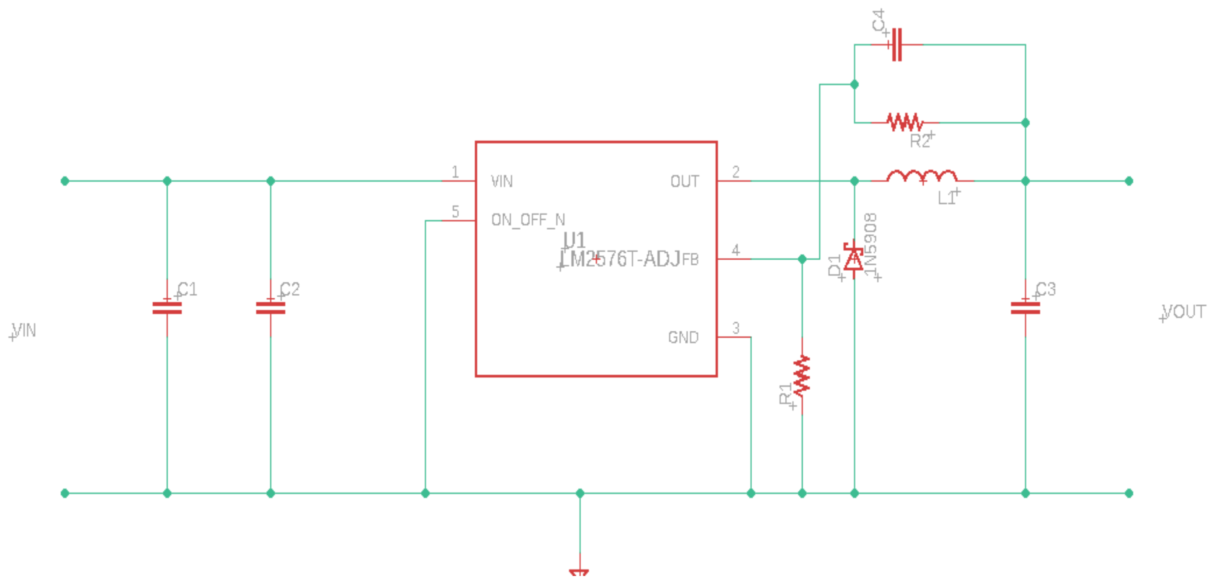


Figure 17: Sample Buck converter Diagram:

This Figure shows one Buck Converter schematic that was created for the purpose of providing power to our MCU based beacons. As mentioned above this buck converter utilizes a Lm-2576 voltage regulator as the main component. The above diagram also shows the typical application of a fixed voltage output converter. This shows the Cin and Cout Capacitors as well as the place of the diode and inductor surrounding the converter integrated circuit.

Section 1: Input Voltage

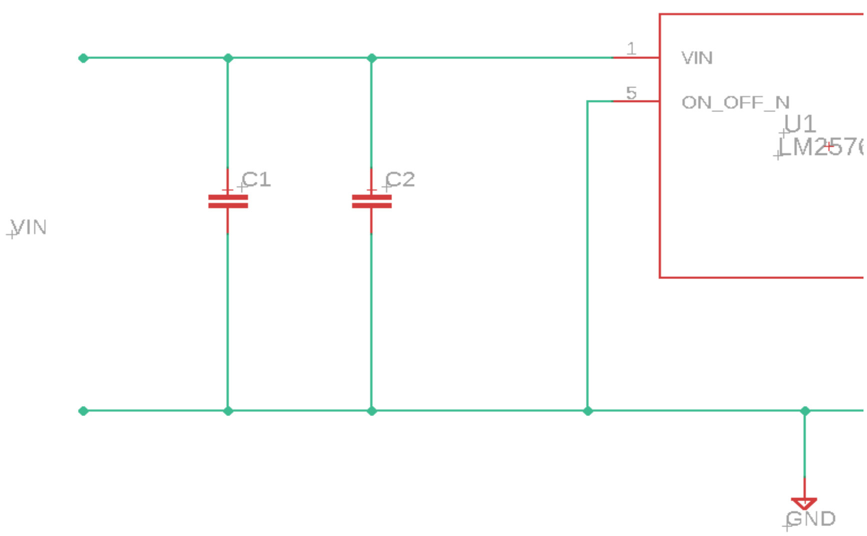


Figure 18: Input Voltage Section of Buck Converter

In this section of the circuit the input voltage for our project will be provided by 5-volt batteries (Vin), and this Vin will cross C1 and C2 where they are an electrolytic and ceramic capacitor respectively. And the values for these capacitors will be **C1= 220μF** and **C2 = 100μF**. From the 2576 we can see that pin 5 (ON/OFF) will be connected straight to ground and will be defaulted to the ON state. The main purpose of these capacitors is to dampen the effect of voltage ripples and noise on the overall function of the circuit.

Section 2: Output

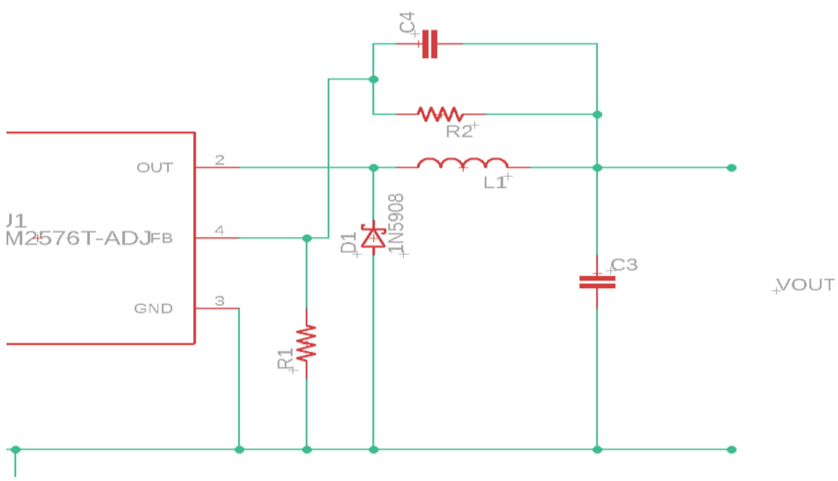


Figure 19: Output Voltage Section of Buck Converter

In the above section of the circuit the output of the buck converter schematic is displayed. PIN 4 the Feedback pin is connected to a resistor, R1 which has a value of **R1 = 330Ω.** This pin is then connected to the top system which includes the **10KΩ** multi-turn potentiometer. This potentiometer is connected in parallel with a C4= 100nF ceramic capacitor and will determine the value of the voltage measured at Vout. There is also a reverse bias Schottky Diode in parallel with **Cout = 220μF.**The Schottky diode is meant to combat any possible voltage spikes due to the inability for current in a conductor to change instantaneously. Basically, protects the circuit from any Inductor load and is a rectifier or “catch” diode.

PCB design #2: Fixed Voltage Regulator(5V)

Another PCB design the group considered was a much simpler fixed voltage power supply. This power supply will take a voltage input (minimum of 7V) and through the use of an LM7805 voltage regulator.

LM-7805 Voltage Regulator

This line of fixed voltage regulators is responsible for taking an input voltage>7V and converting it to an output of 5 volts that we require for our MCUs. This integrated circuit only requires a minimal number of external parts to complete the full buck converter. This regulator allows us to drive the required 3 amps as well and includes safety measures to stop any failures from harming connected devices similar to the previously mentioned LM2576 module.

Circuit Elements:

1x100μF Capacitor
2x100nF Capacitor
1xLM-7805 Regulator

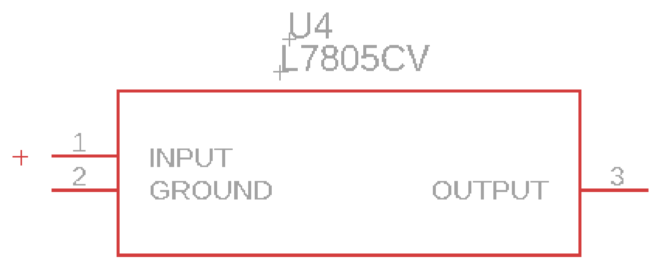


Figure 20: LM-7805 Integrated Circuit diagram

The schematic above is the lm-7805 block diagram and the pins work as follows:

PIN 1, Vin:
The function of this pin is to take the positive input voltage to be converted and send it
PIN 2, GND:
When the integrated circuit completes the step down the voltage is sent out of this pin
PIN 3, Vout:
Standard ground pin of the integrated circuit

Sample Voltage Regulator Diagram:

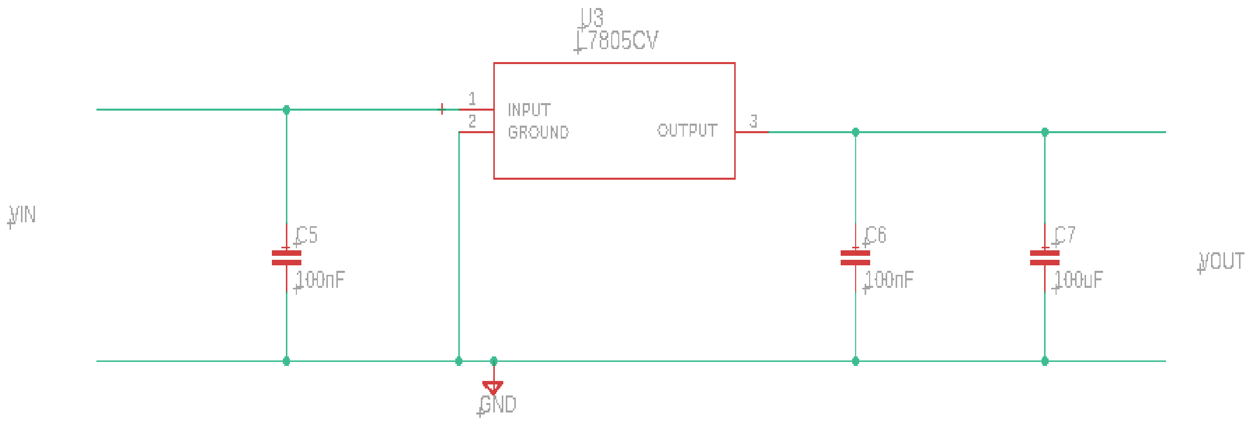


Figure 21: Voltage Regulator Schematic Diagram

This figure above shows another power supply schematic that was created for the purpose of providing power to our MCU based beacons. As mentioned above this voltage regulator circuit makes use of a LM7805 voltage regulator as the main component. The above diagram also shows the typical application of a fixed voltage output converter. This circuit design is much simpler than that of the buck converter and this power supply also has the benefit that the power supply is fixed so in theory there should not be a large variation in the output voltage readings. This simple circuit may need modifications, such as the addition of a noise cancellation/dampening module for the input voltage to clean up any issues with the way input is processed. It contains noise cancellation for the output, however. This implementation requires that we provide at least 7V to the input voltage so we need to make sure that any batteries we use as input will provide at least that much power. It is possible to increase the output voltage of this circuit by implementing 2 Zener diodes.

The input capacitor C1 will have the value of 100nF and will act as a ripple regulator on Vin. capacitors C2 and C3 will have the values of 100nF and 100uF respectively. These capacitors will dampen any excess noise in order to achieve a fixed Vout value.

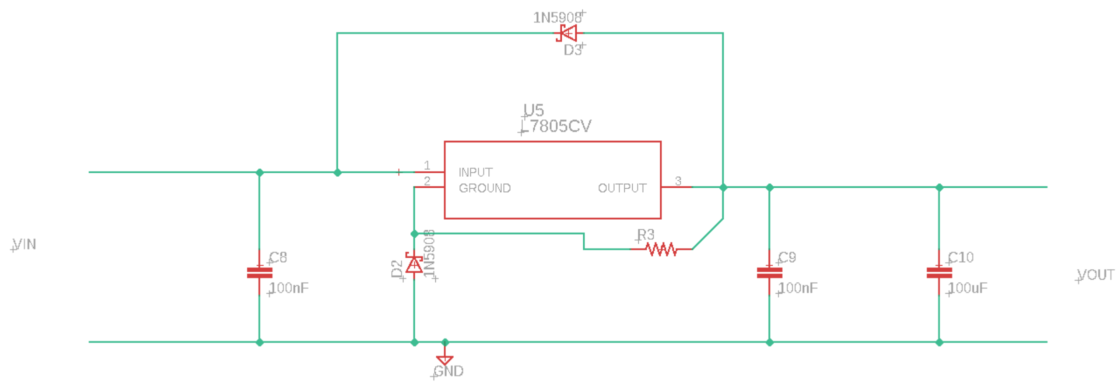


Figure 22: Alternate Voltage Regulator Design

The figure above is a schematic diagram of the design possibility discussed in the previous section. Here we have added 2 Zener diodes and a resistor. As an example, if we wanted to increase our voltage from +5V to +9V what we could do is set **R1= 910Ω.** This value is found by assuming input is 12V and finding that 5V/5.5mA = 910.

The Zener diode D2 requires a reverse bias I of approx. 5.5mA when including current "lost" to the regulator. The feedback diode stops the regulator from being in reverse bias mode when input supply is 0V or non-existent while the output voltage is still in a non-zero state.

This third alternative voltage regulator design can be a useful design to implement in the case that we utilize different Microcontrollers that require varying levels of voltage, since we have been able to use one integrated circuit regulator while being able to design and manufacture more PCBs at different output voltage levels, while only having to change the resistor R1 value. This is also an argument for the Buck converter circuit design. For example, if we do end up purchasing different models of Microcontrollers then with the buck converter, we have been able to order the same PCB and can change the value of the output voltage by altering the resistive value of the potentiometer.

5.1.6.2 Final PCB Design Schematic

Ultimately, our team decided to go a different route with our PCB design. Instead of using the PCB to power on the Jetson Nano, we decided to use it as our beacon and tag components. We opted for this route since the Nvidia Jetson Nano was not operating as expected due to its memory capacity and not being able to run face recognition. For our project we have two generations of design. Our printed circuit boards (PCBs) were designed in Eagle and EasyEDA due to the extensive library available. The microcontroller we selected based on popularity, price, and extensive documentation was the ESP32 WROOM 32D module. The ESP32 module satisfied our requirement for Bluetooth and Wi-Fi components since Bluetooth will be used for the implementation of our indoor tracking and Wi-Fi will be used to send over data from our printed circuit board to our web application.

Originally our first-generation design was made with the intention that it would be a standalone beacon. Our first-generation design was mainly a voltage regulator designed to power on a 30 pin ESP32 development kit module that we had purchased prior for testing our initial code. Due to its relatively large size of (115mm x 67mm) we decided that it would be best for this design to act as just a stationary beacon that will emit Bluetooth signal using the ESP32 module from the development kit, as it would be too big for an individual person to carry around as a tag. The main voltage regulator being used in this design is an LM7805 voltage regulator. The regulator is an old component that turned out to be inefficient. To power on our first generation design we used 12 volts to 9 volts male power jack, which the regulator outputted to 5 volts giving us an efficiency rating around 41% - 55%.

For our second generation design we decided we wanted to essentially create our own development kit and get rid of other features that we do not need. The main component of this design was again the ESP32 chip module. Unlike our other design, which consisted of through-hole components, our second-generation design used surface-mount devices which are smaller, cheaper and more efficient. Our second-generation design consists of three main subsystems, the ESP32 module, the power circuitry for the module and the USB-C circuitry which is used for uploading and pushing code onto our ESP32 module. Our new printed circuit board design can be powered on by either a USB-C input or by connecting a LiPo battery pack, giving it the option to be either stationary as a beacon or mobile as a tag. The LiPo battery is also able to be charged via USB-C connection by the MCP73831T component to increase the lifespan of our beacon/tag printed circuit board design. The surface mounted regulator we used is the XC6220B331MR-G. This regulator is more efficient than our first-generation design, taking in input voltages of 3.7 volts from LiPo and 5 volts from the USB-C connection we see an efficiency rating of 66% to 89% which is a huge increase compared to our first-generation design.

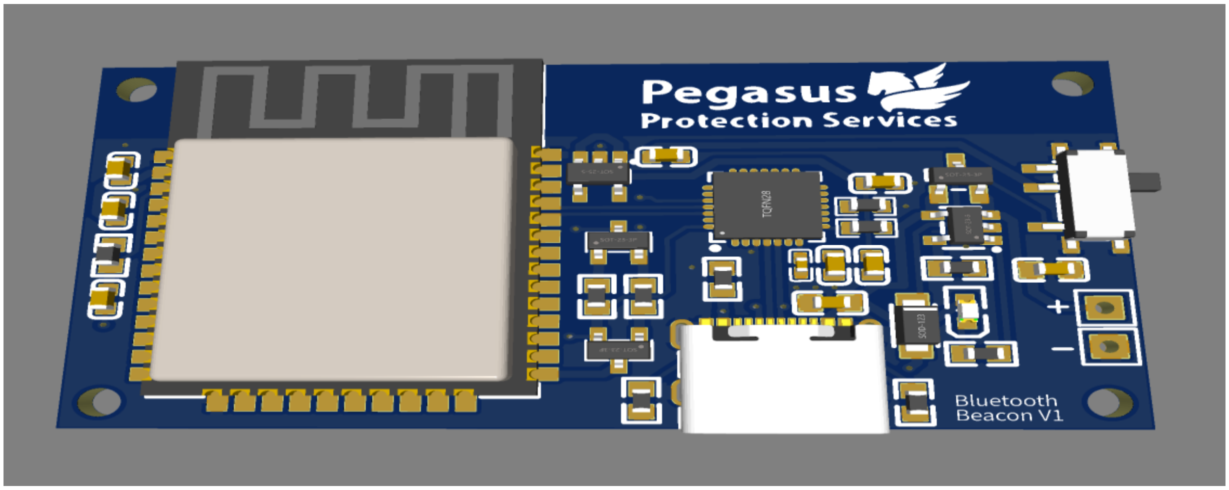


Figure 36. 2nd. generation PCB design for use as BLE Tag/Beacon

The main improvement our second generation had was size. Due to the use of surface mount devices, we were able to reduce the size to 50mm x 28mm x 10mm.As a result, our new design is now able to act as either a beacon or a tag depending on what code we plan to upload on them and whether you want it to be powered on via a regular USB-C connection or LiPo battery. This reduction in size makes it easier for consumers to place around their specialized location(beacon) and easy enough for them to carry around in their pockets(tags). In Figure 36 you can see the finalized design of our second-generation beacon.

The full schematic of our design can see in figure 37 below:

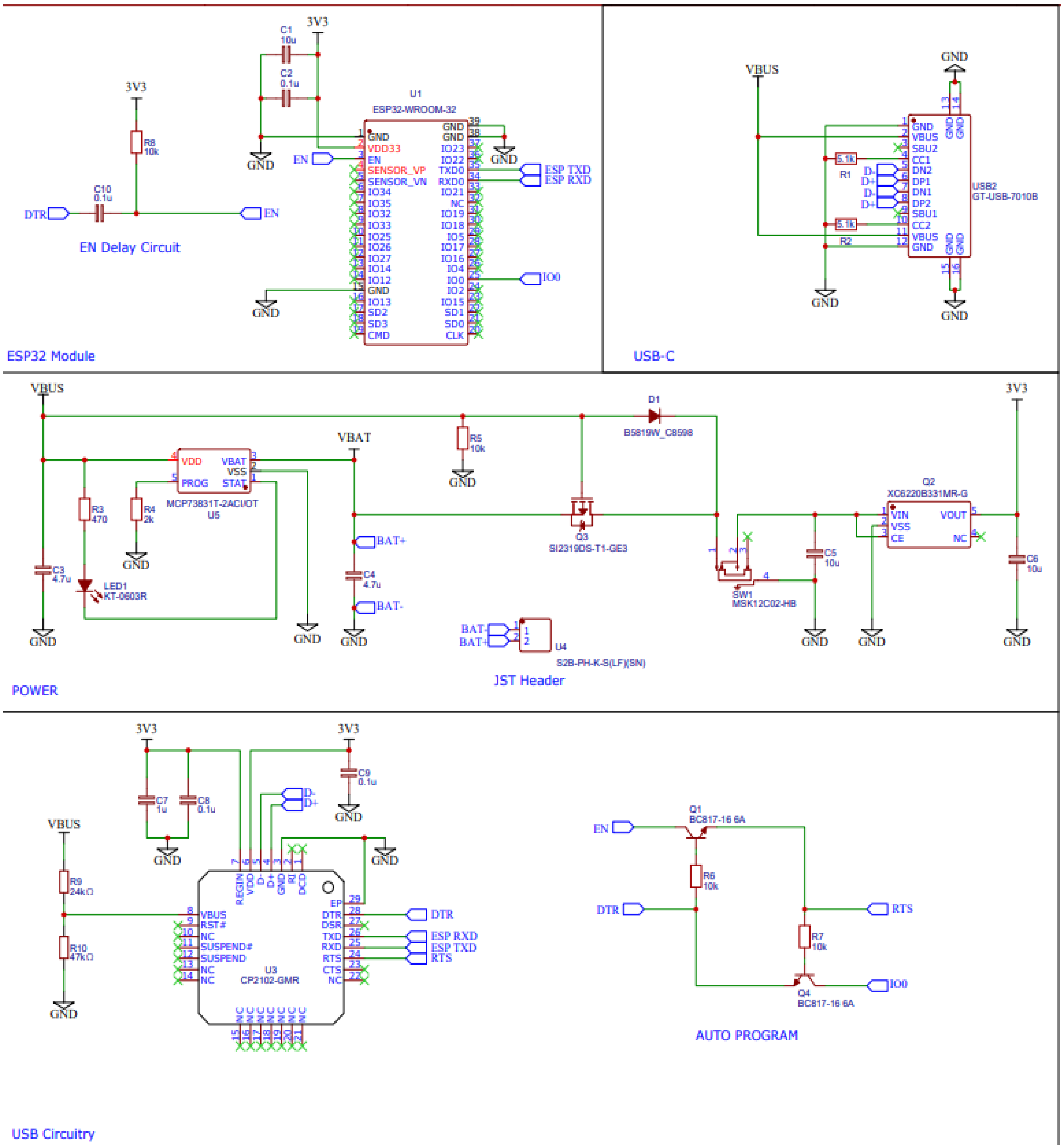


Figure 37: 2nd Gen PCB Schematic

5.2 Software Design

In this following section, the software design will be explored in detail. Software design is a major component for our security system as it lays the foundation for the User interface and the facial recognition system. For the user interface we dive on the frontend design and the API's endpoint we use as well as the database and the mobile application design. When reviewing the facial recognition system, we have touch on the technology we use, and the support vector machines we plan on using. Overall, the following section provides the guideline on the complete user interface and facial recognition system.

5.2.1 Computer Vision Component

At the core, the software side of the security system our group is creating is built widely on the foundation of computer vision and machine learning concepts. Since our Bluetooth card tracking system necessitated a method for handling/catching intruders who are not directly linked with the system, such as an outsider not working at said occupation, our group elected to utilize cameras that will be able to handle when such scenarios occur. Now that we have the devices that will serve as the eyes for our system, where exactly does computer vision come into play? We dive into further detail shortly... but first: What computer are we going to use to perform computer vision within the security system?

The "Computer" At Work

As mentioned in the hardware section of our design, we are employing a microcomputer, the NVIDIA Jetson Nano (The Nano), and designating its job to running programs that will perform computer vision concepts for the entire system. Because we are using OpenCV to develop the program that will build our models for computer vision, we have already set up The Nano with the Python/C++ programming language and OpenCV installed. Arguably, The Nano adequately suits our needs for the system; we needed a computer that is portable, relatively affordable, and comes with sufficient memory that can handle executing programs on the complexity that normal face detection/recognition algorithms run on. It is the crux that effectively implements the computer vision side of our project, and without it, could leave our system vulnerable. With that understood, we have now discussed our method for creating the face detection model our microcomputer will call on to evaluate incoming images from the cameras.

Implementing Face Detection

The first obstacle our cameras pass on to the microcomputer are images that need to be parsed through to determine if a face is located within the image. This process is known as face detection, and work in the computer vision field has been done to sufficiently handle this occurrence. Although algorithms for computer vision are a relatively recent development in the grand scheme of things in the world, there are quite a lot of public resources for implementing a handful of these algorithms that surprisingly perform adequate for most uses. We make use of this observation by using the libraries OpenCV provides to the public to construct our face detection model. Since machine learning models take quite a bit of time to develop, some libraries have been kind enough to provide pre-trained models for computer vision. Here is a simplified overview of the steps we performed at this stage to implement face detection on our microcomputer:

1. Initialize program(s).
2. Import OpenCV packages.
3. Define a new face classifier to observe faces.
4. Provide the model data.
5. Train our own face classifier based on the data.
6. Chain a series of pre-trained models together to serve as a global face classifier.
7. Append our trained model along with the other models.

This overview is solely meant to briefly describe the process we followed. It seems like a walk in the park when you lay it out as such, but there is a considerable amount of work that will need to be done at each stage. We have now shift gears to elaborate on each stage particularly, and by doing so hope that anyone reading this achieves sufficient comprehension of our development for the face detection portion:

Initialize Program(s): We have likely use one or two programs to allow for proper coding/understanding of the face detection component. Ideally speaking, we are going to use one program to build our own face detection model, and once the model becomes fully trained, will we import it into our second program which performs the chaining of our models with other models into one big ‘expert’ that can classify images as having a face or not. As for the language we are using, there are two roads we are exploring. OpenCV is traditionally a library for C++ programming language, but bindings allow for coding in the Python language as well. However, upon observing the provided documentation and demos of code on their website, it seems like C++ is ideal to use as there is simply more documentation on that language.

Import OpenCV packages: The syntax to import packages to use in a program file is not the cumbersome task about this part. Instead, it required us to investigate the OpenCV documentation to see the libraries available for us to use. As per the docs on [35], there are 30 main modules out there, with 32 extra modules to also consider. While we haven’t thoroughly looked through each library available to us, we had some ideas of where to start. Since face detection is a branch of the object detection field in computer vision, it makes sense that we at the very least use the objdetect package. Other libraries that we would likely end up using are the ml, cudafilters, imgcodecs, and core.

Define a new face classifier to observe faces: Within the objdetect package of OpenCV, there are 5 possible classes to create our classifier on. We are opting to use the CascadeClassifier class to create our classifier for which we have train our model on. The other classes such as QRCodeDetector, QRCodeEncoder, and SimilarRects did not seem to fit the objective of our goal in this component. It should be noted that this classifier uses the Haar Feature-based Cascade approach [36] discussed in the Relevant Technologies section of this paper. Below is illustrating the set of Haar features utilized by the algorithm:

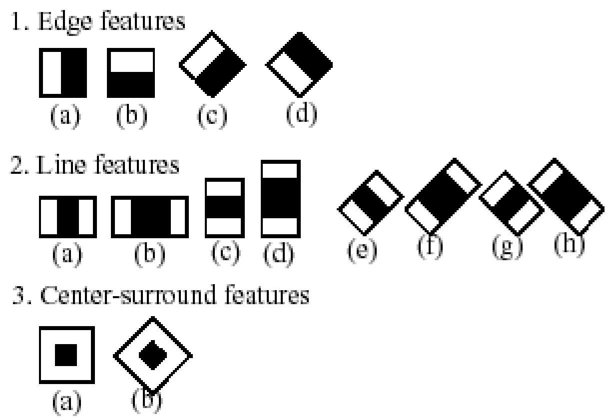


Figure 23: Haar features for CascadeClassifier

Provide the model data: We need to train our model if we want it to become a better classifier, and the only way to do that is by passing data into the model. In reality, the “data” we pass is a collection of positive and negative image examples. We do this so the model is not restricted to only instances where faces are/are not in the image. For our positive image set, we are utilizing a huge dataset known as the Tufts-Face Database. For the negative image set, the only requirement for each image is that it does NOT contain a face in it; even images with not real faces (such as one on a painting on a wall) will not classify as a negative example and may cause difficulties if the model were to train off it. As for what percentage of positive to negative examples we have included in our dataset, we have use 75% of the dataset as positive examples and the remaining as negative examples. The great thing about the Internet is that there are plenty of open resources from which you can grab positive/negative examples and is saving us a long time in the long run from taking the images ourselves.

Train our own face classifier based on the data: Arguably the longest part of this component is training the classifier. In code, it becomes as simple as calling a fit () method on the class, but what the computer begins doing is taking each of the observations it was provided and passing it through as input to the algorithm the model is based on (in this case Haar features) and computing a prediction based on the output of the algorithm. You can imagine these algorithms being quite complex in nature, and as a result will take time to fully learn off each example. If the algorithm were as easy as addition on paper, there would be no reason to employ the work of a computer. We have been using one person's home computer to fit the model as their computer proves strong in computation problems just based on the memory and processing power it has.

Chain a series of pre-trained models together and append our trained model along with the other models: We have been informed that computer vision projects have not worked convincingly in previous years, and to combat this, we plan to create a layering of open-to-the-public models in conjunction with the model we trained to achieve an adequate face detection rate. This strategy is also known as the ensemble method in machine learning, and it is widely used as it generally performs better than using a single model as a predictor. There are a couple ways this ensemble can operate. You can give each model a weight, and the model with the highest weight has more influence than models with a low weight. Another approach would be to simply look at the yes/no prediction between each model used, and the prediction with the most occurrences would be the prediction outputted by the ensemble overall. We are going to use the second approach first, see how it performs, and compare it to the first method. Whichever yields better accuracy is the process that will stay in the final implementation. Example of an ensemble layout:

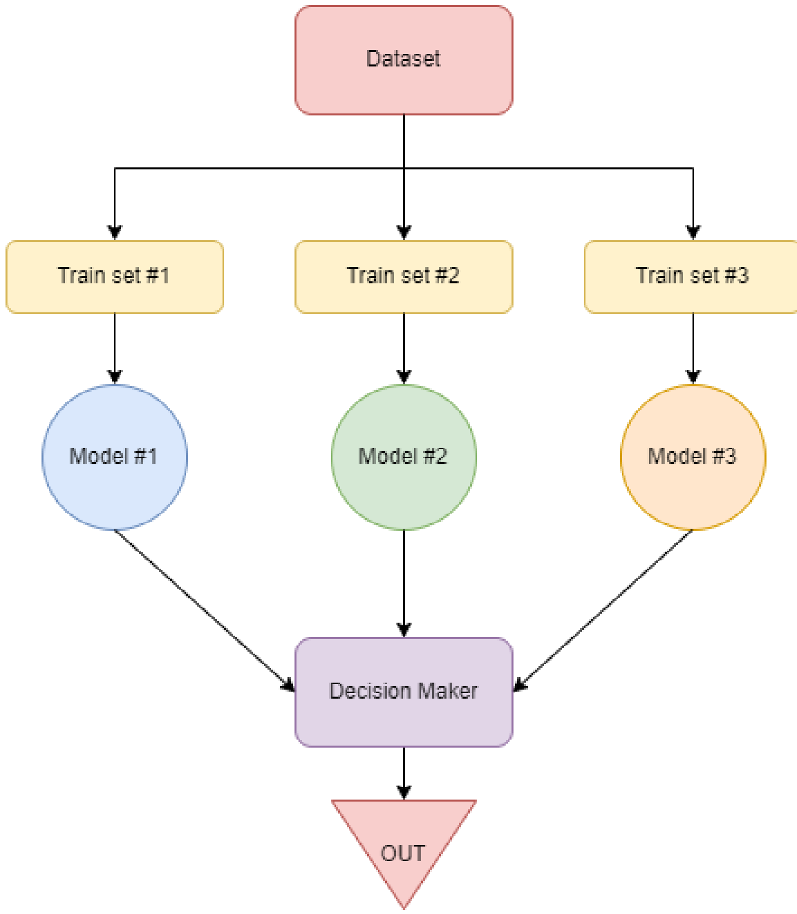


Figure 24: Ensemble Format

Implementing Facial Recognition

Up next, now that The Nano observes a face in the image, is determining whose face is within the image. Again, we needed a model that can point a face in an image and match it to a face that is stored in our database of 'accepted faces'. OpenCV has a couple of various libraries that could be utilized to build a model for this purpose, and so we are opting to use it again as our foundation for the facial recognition component. Here is our general plan for the facial recognition portion of our system:

1. Initialize program(s).
2. Import OpenCV packages.
3. Define a new face classifier to identify faces.
4. Provide the model data.
5. Train our own facial recognition identifier based on the data.
6. Use a series of facial recognition algorithms together to identify faces.
7. Append our trained model along with the other models.

It follows a similar series of steps used in our face detection component, but some specific parts differ, since we are attempting to identify the face in the image, as opposed to a yes/no answer of whether there is a face in the image. The main difference illustrated here is that facial detection is considered a binary classification problem, and facial recognition is a multi-class classification problem. Nevertheless, let us elaborate on what is happening at each step:

Initialize programs: Like the choice made in the face detection component, we are splitting our code up into two programs. The first program will be declaring, training, and tuning a facial recognition classifier provided by an OpenCV library. Once we consider the model to be strong enough, we have then move to creating an ensemble of facial recognition models in the second program that will serve as our main classifier for the facial recognition implementation. Additionally, we may use a third program that will filter/enhance the incoming images from the camera to ensure that our model(s) can interpret the image as clearly as possible.

Import necessary OpenCV packages: After careful evaluation, we believe the proper module to achieve our face identifying needs will come from the DNN (Deep Neural Network) package. This allows us to create a neural network that can become capable of identifying faces after learning from training data.

Define a new face classifier to identify faces: Opting to use a neural network as a classifier greatly differs from using other machine learning models. In the documentation for DNN [37], the primary class we work with is the Layer class. Neural networks operate based on layers, and therefore our model will be using a collection of layer classes to learn from the data. Below is an illustration that provides a general idea of how a neural network is composed:

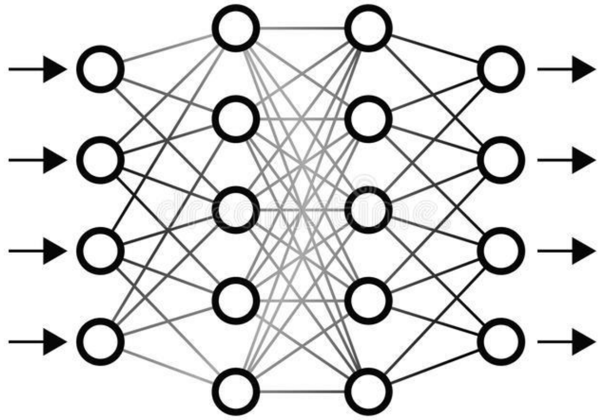


Figure 25: Neural Network Structure

Provide the model data: As opposed to finding random face images from the Internet, we need to provide face images of the people who are 'accepted' as part of the system. It is crucial that our set contains at least 10-20 sample faces per person, that way the network can properly learn the unique features that come with each person's face. There should be a set of non-examples that can be used to verify whether or not the model is learning correctly.

Train our own facial recognition identifier based on the data: Again, the model will likely come with a fit () function that will begin training the neural network when it is called based on the data it is being asked to fit. We are aiming to use a maximum of 1000 epochs for the model, and as a result the model will take a considerable amount of time to train. We have again use one team member's powerful computer to expedite the training process. By the end of the training process, the network should have learned the features that make up the 'accepted' faces in the database and is ready to serve as a predictor alongside other models.

Chain a series of pre-trained models together and append our trained model along with the other models: Finally, we wrap all models into one overseeing model that will aim to identify faces incoming from the cameras.

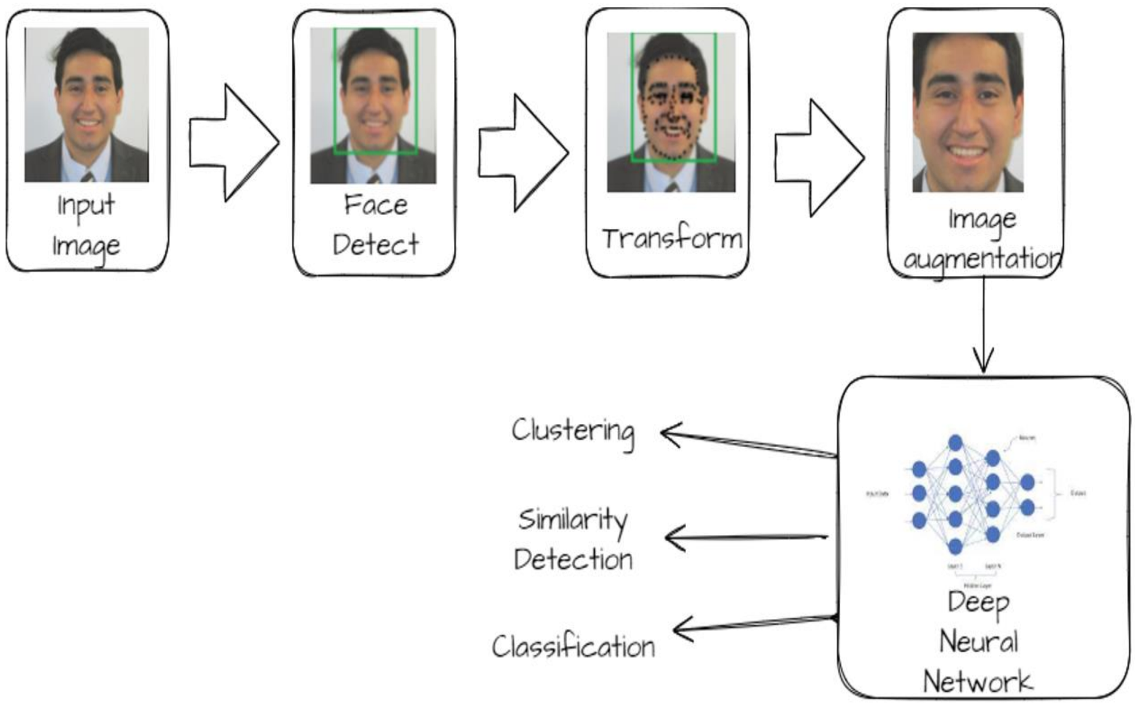


Figure 26: Overview of Computer Vision Component

When The System Raises a Flag

Now that we have detailed the process for implementing our face detection and facial recognition components, how does the system use these to detect an intruder? What happens when such a situation occurs? This can be determined within one main overhead program. Only a few cases will trigger a warning that is passed to the administrators of the system:

1. Face is detected but could not be found in the 'accepted faces' database, either due to it being an intruder or error made by algorithm.
2. Face is detected, it is valid in the database, but in a location that they should not be able to access.

Since most of the heavy computational work is done when the face in the image is being mapped to a face in the database, all that's left is to determine what location the face was detected in. The cameras will work with the beacons around the building to send this information to us.

This section has essentially described our methodology for implementing a working tracking system, but what about the scenarios when the system misidentifies a face as an intruder? Can an intruder pass the ‘face recognition’ test and be misinterpreted as a valid face in the system? We have now dive into both cases should they occur in the next two sections below.

False Positives

There are always cases to consider if a system fails to operate as intended, and in the case of our tracking system, false positives are included as one of the cases. Put simply, false positives are when something gets marked as valid/correct when in reality it is invalid/incorrect. For our computer vision component, this happens when a face in the image sent by the camera has a face that is improperly identified as one that is in the ‘accepted faces’ database. The computer handling the computer vision will report no error when this occurs, meaning the ‘intruder’ will successfully bypass security until they encounter another security ‘gate’. How does this happen? While there are no certain reasonings behind false positives occurring in our systems, there are some scenarios that could help explain why it might happen. For example, since facial recognition often relies on facial features of a face to determine the identity of the face it is looking at, perhaps an ‘intruder’ closely matches the identity of someone who works at the building. It may make more sense if that ‘intruder’ was later revealed to be a close relative to an active employee. Lighting and/or the angle at which the face is seen could also factor into the times when faces unexpectedly pass the facial recognition algorithm. The team will look to account for these factors when we use the system in a testing environment and make clear of such accommodations in the document. Nevertheless, let’s look at the other possibility for where the errors may occur false negatives.

False Negatives

The other case would be when the cameras signal warning on a face that was detected, recognizing it as an ‘intruder’, and not in the ‘accepted faces’ database, when in reality, the face is that of someone who works there and should be ‘accepted’. As opposed to what happens for other accepted faces, the computer will throw a warning to an administrator, and may cause some problems for the person who simply meant to be doing their job. Again, we can only look at possibilities for why these situations may occur. One possibility is that the ‘face’ in question may be wearing accessories, such as sunglasses, headbands, or a face mask. First two are probably less likely to occur at an indoor workplace, but the ongoing pandemic has made the third more prevalent, for safety reasons. It is true; masks have created another realm of difficulty that current computer vision face recognition algorithms have to tackle. Because the wearing of masks on the face often requires a person to cover their chin, mouth, and part of their nose, the algorithm loses some facial features it could use to help better identify a person. We have put our model to the test with each of these accessories to see how much of a problem it could have on the overall accuracy. Nevertheless, all three could be plausible reasons for marking an actual employee as an intruder in the system. Just like how some of the false positives can be attributed to variable lighting/angle conditions the camera experiences, the false negatives may occur due to similar varying circumstances.

Countermeasures To Errors

First, we want to make clear that while we explore the conditions in which our system will operate its greatest on, there should be no expectation or belief that with ideal conditions in place for the system should yield no errors in evaluation. Computers and even the robot vision field have been around for a relatively short amount of time, meaning that humans are still discovering ways to improve these existing components to make them more reliable and robust. After this is acknowledged, we can begin to consider possible methods for reducing the overall errors that could occur in our system.

Since we have not built a proper testing environment for our tracking system at the moment, it makes testing different conditions hard to qualify and therefore currently we have simply brainstormed some possible avenues in this discussion that, when our testing environment is finally built, we have gone back and redescribe with actual conditions that showed to affect the performance of our system in our testing environment. The angle of the camera at each checkpoint in our system we believe needs to be consistent in a manner that allows for the camera to observe a face straight on over 80% of the time. The simple reason for this is that our model will be built and trained on faces that are from the same angle, so if it were to observe a face at other angles, it may have difficulty calculating an output for those other angles. The other condition that we'd like to impose is one that keeps lighting in the workplace that satisfies the typical 10-20 lumens per square foot rule of thumb for room lighting. There is a strong possibility this will change as more testing is done when the testing environment is finished being built.

5.2.2 Database

In this section we cover in depth the details of the database and how we plan on keeping tracking of users, location and the recognition system. We'll need multiple databases for a few components in our project, which include:

1. Database for Facial Recognition
2. Database for Tracking System
3. Database for Application Credentials
4. Database for Storing Camera Recordings

5.2.2.1 Database for Facial Recognition

For the database framework, since we decided to use MERN Stack as our primary technology set for this project, we have been using the non - SQL database known as MongoDB. MongoDB is a highly scalable and flexible database that uses a JSON-like format to store data (https://www.mongodb.com/why-use-mongodb). The database will be built to accept new users. Each user will have a Recognition table assigned to them in a one-to-many relationship table. The recognition table is able to accept and remove any new “target” that the user wants to monitor via the camera system. Once a target is added in the recognition database by an image, the camera system will learn to track and try to monitor a given target. Figure 27 below displays the entity-relationship diagram. This diagram shows the basic understanding of our database will work.

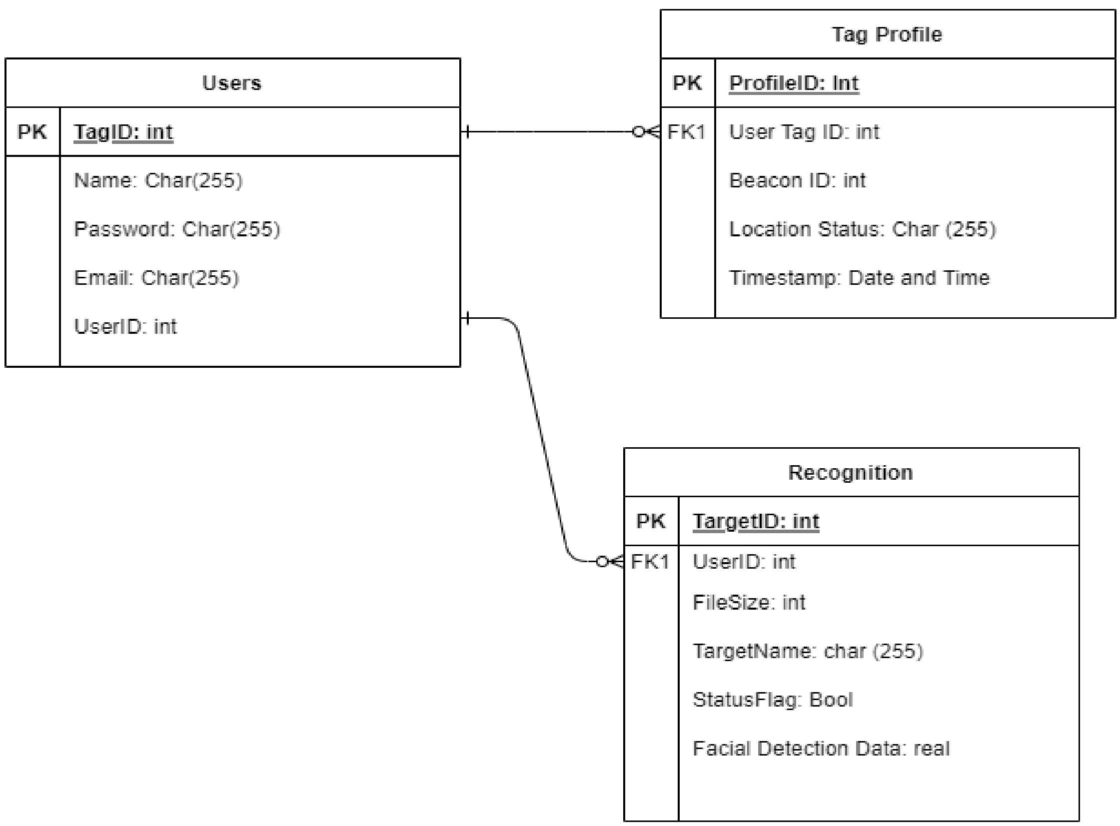


Figure 27: Entity Relationship Diagram

The lines in the diagram above represent its relationship with each table. As you can see the User, at creation would be given a Bluetooth tag with an ID on them, this ID will be the primary key to connect it to the Tag Profile table. The primary key acts as an identifier. So, each location is based on the tag. The four-beacon system will emit signals for the tag to receive, once the tag is detected within the area, the database will create a record in the Tag Profile table. Each record will be inputted with a specific user Tag Id, which will map accordingly to the TagID in the user table. This means that users will be able to gather tracking information via the tag they are given. You will also notice the recognition table which will be responsible for creating new input for our facial recognition system. The way it works is that users are going to be able to upload images of individuals they want the camera to recognize. So, consider this scenario, if you are the security admin for a company, you will have to upload images of employees that you want the camera to be able to detect. Once images have been uploaded, they will be sent to the facial recognition software for training via the backend APIs. Next, we have gone into more detail about the indoor tracking database and how records will be created.

5.2.2.2 Database for Indoor Tracking System

This is a sample database and the information here is subject to change. For understanding consider the following scenario, there are two microcontrollers and four beacons in our indoor position tracking system. The layout of Engineering II Atrium features a set of computers, a rocket ship, and a winding stairwell, this is the database that will be for the users and their corresponding tag.

MCU_ID	Beacon_ID	Location	Timestamp
1	10	Rocket	2017-07-04 13:23:55
2	11	Computers	2017-07-04 13:23:55
1	12	Stairs	2017-07-04 13:23:55
1	13	Computers	2017-07-04 13:23:55
2	10	Rocket	2017-07-04 13:23:55
1	11	Stairs	2017-07-04 13:23:55
2	12	Rocket	2017-07-04 13:23:55

Table 10: Location History Table

5.2.2.3 Database for Application Credentials

The application will also need a database to hold credentials of users that are recognized as administrators and therefore should have access to monitoring the information provided in our system. This can be achieved using either SQL or NoSQL language to hold credentials of administrators. A collection of users with their personal information (username, password, email, first name, and last name) will be recorded to uniquely identify each administrator in the application. Here is a simple example of the collection that will be used to store login credentials:

ID	Username	Password	Email	First Name	Last Name
0	chri5t	####	cs@outlook.com	Christian	Silva
1	dsau	#########	ds0715@gmail.com	Dylan	Sauerbrun
2	rem1	########	remwil@yahoo.com	Remi	Williams
3	aunf	######	afred@aol.com	Aundre	Fredericks

Table 11: Users Table

5.2.2.3.4 Database for Camera Recordings

Another database that is relevant in this discussion will be the database that holds the film for the last two days of recordings for each camera. This will likely be a cloud platform that we can find either for free or at a relatively inexpensive rate since we'll need to keep history for each of the cameras in the tracking system.

5.2.3 Mobile and Web Application

For the web and mobile application, we want to create something that will be easy to use for those who implement our system into their buildings/location. Since the information they will see is proprietary, only selected employees will be allowed to make an account - depending on the scenario most security groups and building managers will have access to this application. We plan on following basic software regulations of creating a login and register page which can be seen below in figure 28 and figure 29. These pages will be the first things users will see when they access the application:

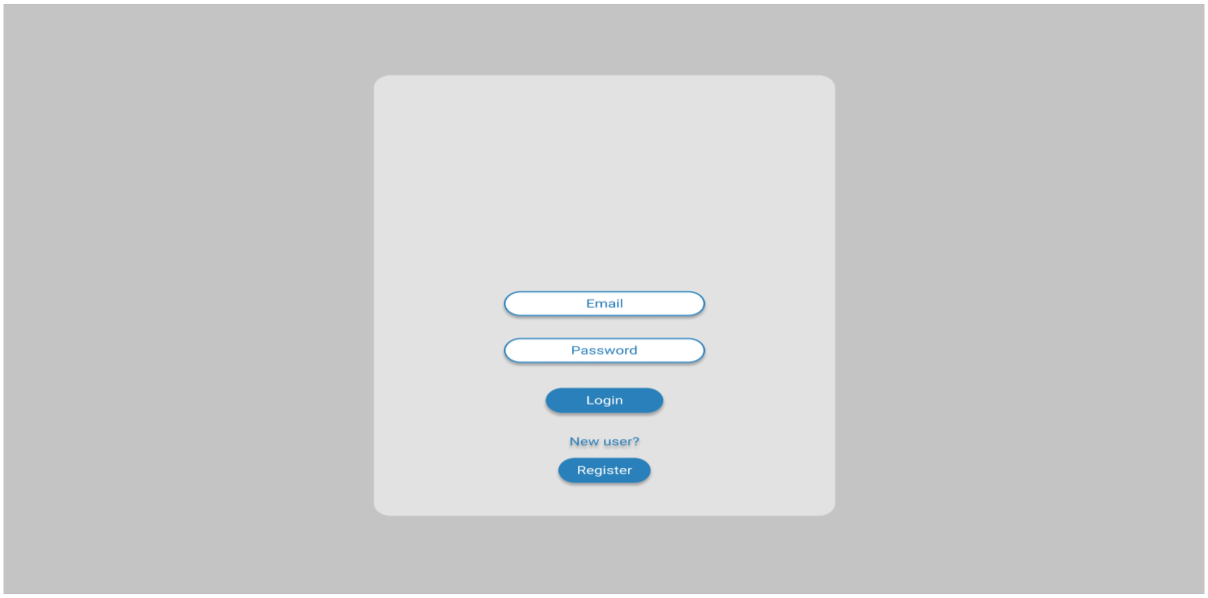


Figure 28: sample Login Screen

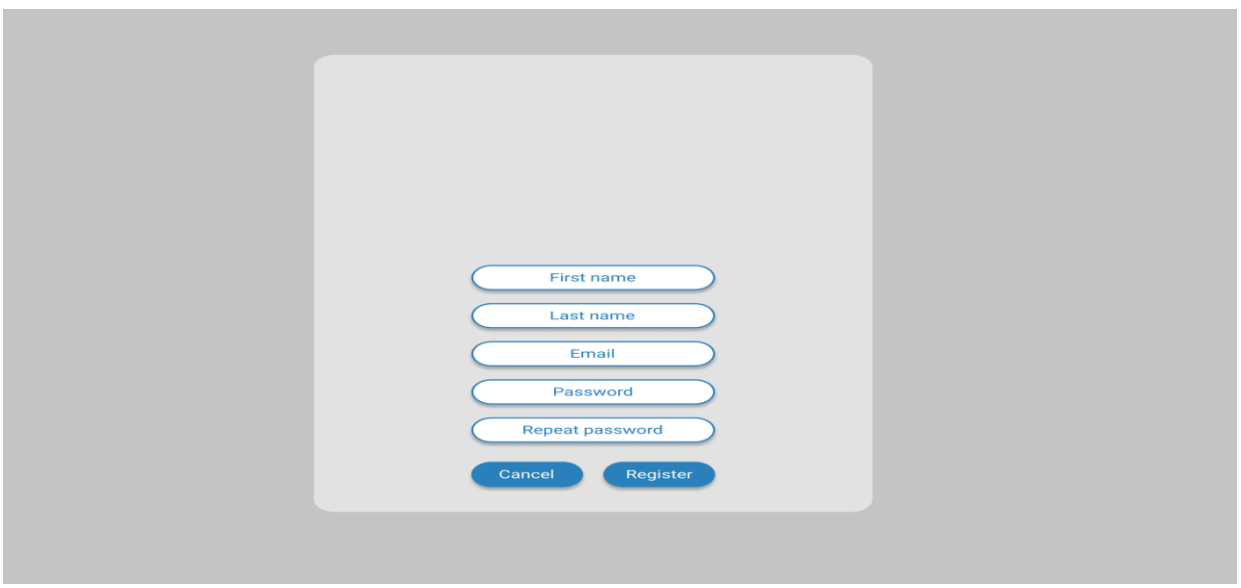


Figure 29: sample Register Screen

When users are logged in, they will be directed to the home screen. The home screen will have options to update users' individual profiles in the settings tab. They will have a tab dedicated to uploading images of individuals for the facial detection software to train and learn to recognize. Most importantly on the home page they will see a table which keeps track of all the tags of current employees or current individuals who have access to the building. The table will display the TagID, the user's name and the room location they have entered at the corresponding time stamps. Since this project is self-funded, we have only had 4 beacons enough to set up two small rooms, so the room location will either be room 1 or 2 with the current timestamp of date and time. The beacons will be placed in only secure areas not the whole building. Below you will see an example of our home page rough draft:

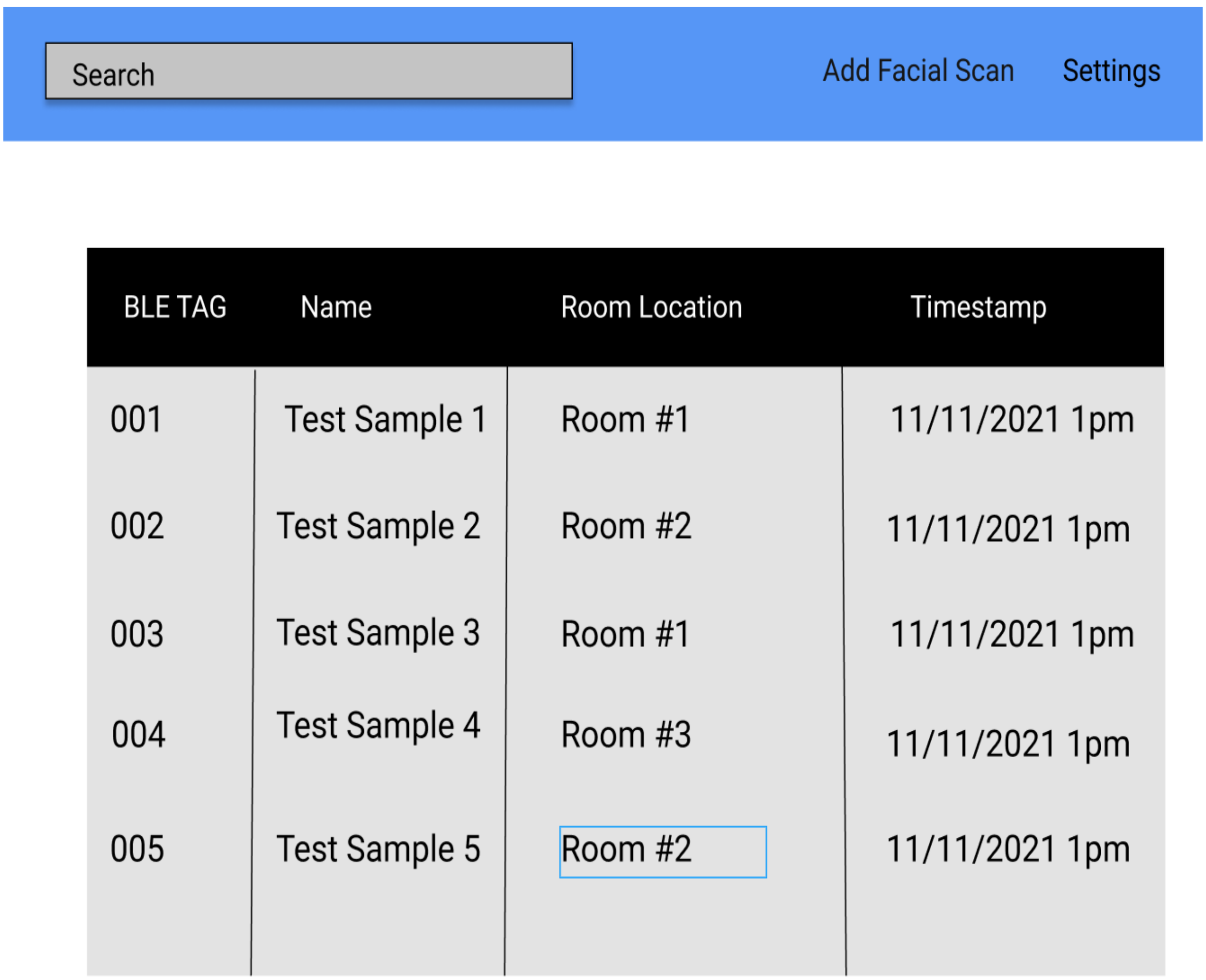


Figure 30: Sample UI Design

Development of the frontend for the web and mobile application will be mirrored via bootstrap, which basically ratios the screen based on the resolution of the device you are using. Both versions will have the same functionality and look to them.

Our application will have various endpoint application programming interfaces, otherwise known as API's. The class diagram below in figure 31 shows the entities, attributes, associations and functions for users and their functionality. The resulting diagram shows visually the plan for the API's being used.

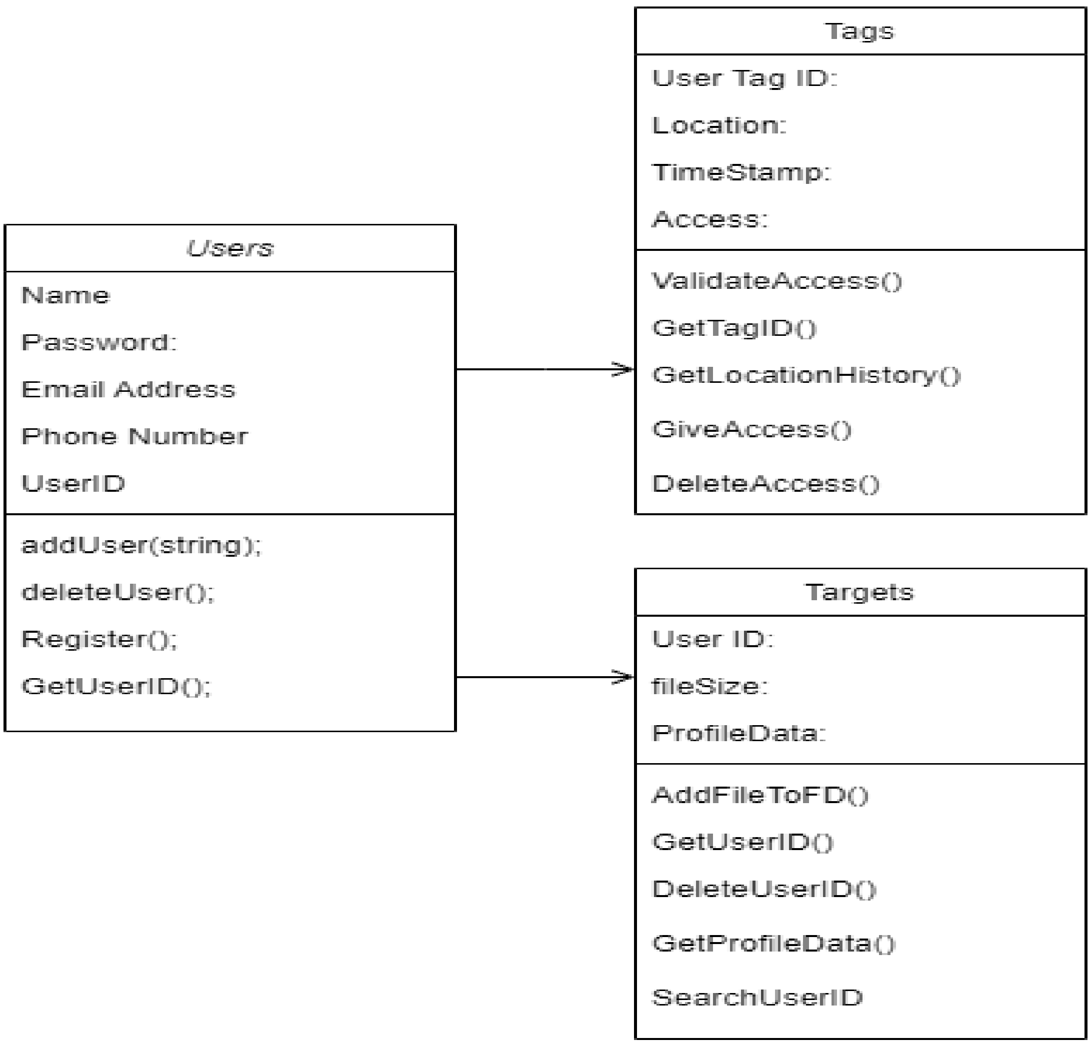


Figure 31: UML Class Diagram

You can from the diagram the various methods we plan on incorporating into our application. Various CRUD functions (Create, Retrieve, Update, Delete) will be used for our web and mobile app to get and manipulate the information within the database using scripting languages, in our case JavaScript.

6 PROJECT CONSTRUCTION AND CODING

As our team continues to create new requirements and identify our design as a whole more, it is necessary to assemble the various subsystems and create prototypes. By creating and building out these prototypes we gain a better understanding of how each system functions separately before putting everything together to ensure that it works as expected. Although the beacon system and the camera system will not communicate with each other, they both work together to ensure that security is the main priority. Given that we are discussing the subsystem, it is important to note all the working components, along with the discussion of subcontractors and suppliers that will be utilized for our project. This section will discuss our prototype systems along with bill of materials and our plan for the suppliers and PCB vendor and assembly.

6.1 Bill of Materials

The smart BLE security system requires a variety of components for proper functionality. Throughout the project, various parts will be acquired from several online vendors depending on availability. As stated before, due to covid-19 there has been supply shortages for chips and products as a whole, so our team has conducted deep searches to obtain these products. Although we have a general idea of all the products we need, there is a future situation where we would have to update this bill of materials for unforeseen circumstances that arise in the future. Below in table 12 is a list of expected parts.

Item	Quantity	Status
Raspberry Pi HD Camera	1	Received
Jetson Nano (2GB)	1	Received
Jetson Nano Battery Pack	1	Received
Arducam Pan-Tilt Platform	1	Received
Single Board Microcontroller	1	Received
Circuit Elements (Resistor,	1-10	Received

wires, etc.)		
Dialog DA14531 BLE Beacon	3	Received
Dialog DA14531 Asset Tag Card	1	Received
Jetson Nano SD Card	1	Received
PCB Board (Custom)	10	Received

Table 12: Bill of Materials

6.1.1 Suppliers

To gather the components that were listed and shown in the bill of materials, our team purchased and planned on using various vendors that are reliable and well established. Our team searched for the best possible price to make this project as cheap as possible due to the fact that this project is being funded by our team itself. Table 13 below displays all the various suppliers that we plan to make purchases from.

Supplier	**Type of Merchandise**
UCF ECE Department	Circuit Components
Texas Instruments	Circuit Components
Amazon	Misc.
Arducam	Servos
Feasy	Beacon/Tags

Table 13: Suppliers Table

6.2 PCB Vendor and Assembly

During Senior design 2 our group will finalize all designs and start construction on our devices. Some aspects of our hardware setup can be put together by our group, for example if the need arises, we have been able to create a BLE beacon circuit, but more intricate manufacturing will need to be outsourced to third party companies. As an example, we have been creating a power supply PCB for our BLE beacons and to successfully complete this we made use of contractors who assemble and manufacture PCBs. Utilizing their services allow us to focus on the parts of the project that are within reason and scope for us to do on our own. In this section we discuss the various 3rd party contractors we have considered and will discuss their pricing, turnaround time, and quality among other aspects.

6.2.1 PCB Manufacturing

As mentioned in the previous section most of our construction will begin after we have finalized our designs in Senior Design 2. This finalization will need to be completed as quickly as possible, because we anticipate that there will be some problems, whether it be in shipping delays (due to semiconductor shortage), PCB failures, broken hardware. In order to account for these problems, we have need to plan far in advance to have and test all of the hardware as quickly as possible. The process of fabrication means that the company will make the printed circuit board and nothing more (no soldering of any components).

For designing the PCBs our group’s chosen software is AutoDesk EAGLE, and from this we can get a schematics, board, file and bill of materials, but in order to get the PCB design manufactured our group will need to provide data about the PCB in the form of Gerber files. These Gerber files provide the PCB manufacturer with information concerning the physical makeup of the board and allows them to make it to our specifications. Some of the included pieces of info are the bill of materials, width x length, hole locations, etc.

When considering the third-party companies, we wanted to manufacture our PCBs the biggest factor of this was the cost, as the quality of the popular PCB vendors is generally similar. Our PCB design for this security system is not terribly complex and for that reason we can see generally lower numbers per PCB manufactured. This is important as we have been ordering many more PCBs than what is required in order to be prepared for any mishaps in production.

The first PCB manufacturing company the group considered was PCBWay. PCBWay is a China-based manufacturing company that provides both manufacturing and assembly services. To get an estimate we used the physical specification of our fixed output power supply PCB circuit and found that with shipping included we are able to get 10 PCBs (non-assembled) for approx. $26. Most of this cost is from shipping as the company is based overseas so for this reason, we have also looked at domestic options for PCB manufacturing. The PCBs have a turnaround time of about 2-4 days so we would receive them relatively quickly, which gives ample time for us to test them.

The next company considered was MacroFab which is a Houston-based manufacturing company. By all accounts MacroFab has excellent customer-service which we value highly, considering we are new to this process. Their pricing is competitive and being domestic their shipping times are much quicker than that of overseas manufacturing companies. Like PCBWay they also offer Assembly services which will come in handy if we want to contain our business to one company. A positive of choosing to work with a domestic company is that the shipping prices will be greatly reduced. For example, with PCBWay the cost of the PCBs was only $5, and the shipping was about $20. The price for these PCBs is <$10 as our PCBs are simple and we can therefore order a number of them.

The next manufacturer considered was Advanced Circuits 4PCB, which is an American-based full-service manufacturer. A positive of this manufacturer is that they provided instant quotes for PCBs based on your board's layout, which is good because we have not had to go through the company sales team to begin our comparisons. A big benefit of working with 4PCB is that they have dedicated student programs, in which they provide students with discounts on PCB manufacturing and assembly services. Like other domestic companies we have not had to pay international shipping prices and their costs for actual PCBs are competitive with those of the other companies we've reviewed. The company also provides a scan of our group's Gerber files that will check to see if the PCB can be manufactured and also the price if it can be.

6.2.2 PCB Component Assembly

The previous section was focused on the manufacturing aspect of PCB design, so in this section we discuss PCB assembly. Assembly differs from manufacturing, because in the PCB assembly step the PCB components are actually soldered onto the PCB board. In theory we could do the assembly ourselves by soldering the individual components onto the board, but we feel as if this is not a good use of our time. Soldering many of these components requires a high level of mechanical skill and for that reason we feel it is safer, and more effective to allow a third-party company to do the assembly as well as manufacturing service. All three of the companies mentioned in the previous section offer an assembly service in addition to their manufacturing service, so we feel that it is best to keep both manufacturing and assembly contained to one company. In this section the company's assembly services will be compared.

PCBWay offers component assembly services in addition to its manufacturing. The price for this on their website states that it will cost $30 in addition to the manufacturing as long as it is within 1-20 components. This discount is temporary, so we have been watching for it in senior design 2 as this price falls well within our budget. The pricing is dependent on both the number of PCBs as well as the number of unique components on the PCB. Once again, the fact that the company is overseas means that shipping costs are much higher than that of domestic companies, but the assembly cost discount is a massive benefit. PCBWay also offers good resources for determining price of the PCB before purchase and they also include a Gerber file scanner to get an accurate to the cent pricing of assembly, manufacturing as well as shipping.

MacroFab is the previously mentioned Houston-based company that also provides assembly services. Obtaining the prices for MacroFabs assembly services was not as straightforward as it is necessary to talk to a salesperson. The quality of their designs is good nonetheless and they remain a competitive choice for us when determining the company that we use to assemble our PCBs.

Finally, 4PCB's inclusion of an assembly service makes them a top choice for our group as both the manufacturing and assembly can be done with them, under a student discount price. In addition to this they are based in America so shipping costs and times are greatly reduced from those in other countries. 4PCB gives a few options in the way that we can get the circuit created, for example we can either have them supply all of the components, or we can acquire them and send them or a hybrid of the two where we send certain circuit components to them, and they assemble it for us and send it all back. The low pricing and flexibility of 4PCB makes them one of the leading choices for our group going into senior design 2.

Ultimately, we chose to go with a different provider - JLBPCB. JLBPCB was recommended to use by Professor Richie, as they are a popular PCB manufacturer that previous students have used in the past and delivered quality work. JLBPCB was used to order our first generation and our second-generation design.

6.3 Consultants

The work for this project is to be completed by this team for this project. Our team has access to an abundance of resources from the College of Engineering and Computer Science at the University of Central Florida. Our team was also lucky enough to have a variety of consultants that we can reach out to, those being past and current professors. We did not necessarily have a mentor or an academic advisor for our projects, but we did have a professor who guided us towards the right direction when we reached out. We also knew that if we ever got stuck on a certain aspect of our project that we could easily reach out to a faculty member who specializes in that area to guide us. Below we go over some consultants our group can reach out to guide us for this project.

Dr. Samuel M Ritchie is a professor at the University of Central Florida who focuses on classes such as Junior Design and Senior Design, he is responsible for preparing students for the last two semesters and guiding them to use the proper technology for their projects. We have reached out to him a number of times for his opinions on certain aspects on our projects, for example, our group initially decided to use RFID for this indoor tracking system, he told us that RFID is too simplistic and would not give us what we are looking for and then directed us towards the use of Bluetooth technology. Dr. Ritchie also has provided us with various options on how and where we can incorporate our PCB requirement for senior design. Dr. Ritchie has been making significant contributions to our team throughout this first semester and we know that if we ever get stuck or need help with our PCB design, he is the one to reach out to for assistance.

Dr. Niels Da Vitoria Lobo is a professor at the University of Central Florida who teaches classes based on computer vision. Computer vision is a huge topic for our team as one of our subsystems completely depends on it. More specifically, both the face detection and face recognition component in our software stems from concepts studied in the class. Some individuals in our groups have taken Dr. Lobo's Robot Vision course before and have been using some prior notes and resources to guide us in how we want to model our facial detection system. One of our group members had been provided useful resources for achieving face recognition after speaking to the professor in office hours. While we have not been in thorough contact with him for this project, we know that we can reach out if we ever get stuck or need some opinion so he can guide us towards the right path.

6.4 Hardware Prototype

As our team acquires the necessary parts and components we need for our project, it is important to test each individual parts to make sure they are all working properly, if not we either have to troubleshoot the issues or return the component and order a new one. On top of this, we have two main systems that consist of hardware, the Beacon-Tag system and the Camera Servo System. The Beacon and Tag system do not need to be assembled from the ground up as we are purchasing those components completed and built already, our responsibility is to make sure that they connect and detect each other. Also, we have to make sure that the beacons are easily placed in rooms and that nothing obstructs the signals that they emit. For our Camera-Servo system we have to ensure that first the Raspberry Pi Camera is about to be mounted on the servo without dragging the servo platform down. Then we must test that the servo is still able to move with the camera mounted on top. Prototyping these systems will ensure that our project is ready to move in the right step forward and allow us to start the programming aspect and put everything together.

6.4.1 Beacon Tag Subsystem

The Beacon Tag Subsystem is one of two main hardware systems for our project. This system is primarily responsible for the indoor tracking aspect of our projects as beacons will be placed around a specific room and will try to detect its corresponding tag. The beacon and tags will be made from our 2nd Generation PCB. The beacons can be powered on via a USB-C connection or a LiPo battery and are to be placed around the location that the user wants to track. Once placed, a tag, which is also our second-generation PCB design powered on by a LiPo battery will begin to get detected and send RSSI value to all 3 of the beacons setup around the room respectively.

6.4.2 Camera Servo Subsystem

Prototyping the Camera-Servo system should not be as intense as the beacon system due to system over ability to be constructed/coded by the Nvidia Jetson Nano Microcomputer. Again, we have to test each individual component for our system starting with our microcomputer, we have to ensure that the peripherals are working properly and that it is able to run our code which will be written in Python and C++. After our initial microcomputer debugging it is essential to test the servo platform, which is compatible with the Jetson Nano. The servos will be connected to the jetson nano using a control board and its TX/RX pins, as well as the ground line which will run the motors functionality. The camera of course will also be tested by connecting to the jetson via the CSI Camera input ribbon as seen in the image below Figure 32. Once connected we use a series of Jetson Nano step by step guides to test that the camera is working as expected.

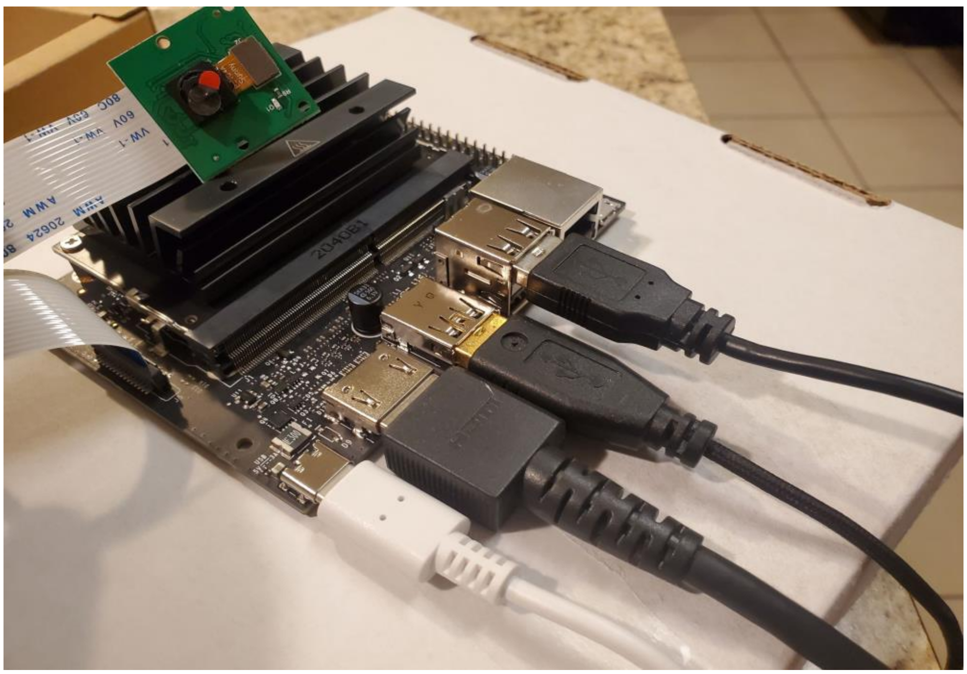


Figure 32: Completed Camera Mount Prototype

When every component is ensured to be working properly, it will be time to prototype the full schematic and connect the components together. The Servo will be connected to the Jetson via the control board and the camera will be mounted on the servo platform while connected to the Jetson via the CSI input ribbon. When the PCB construction is done it will be used to power the Jetson Nano. Once intact the system should be able to move the camera around using the servo platform which will be encoded using our base code via the Jetson Nano, our main goal is to ensure that it is getting the required 3.3V needed to function properly.

6.5 Final Coding Scheme

With consideration of previous sections regarding the approaches for face detection, face recognition, and simply object detection, as well as the Computer Vision Component that describes the process for implementing such technologies, this portion will be sharing the final decisions we have made for designing the code for both the computer vision and partner application components of our project. We hope that by including a well-thought-out plan before the second course starts gives us some direction to fall back on in the event that we get lost during the design process.

The foundation for any coding problem requires one to sit down and draft some pseudo-code/illustrations before typing up any code as a solution to the task at hand. Throughout our years of computer science courses, we have learned that diving into the problem without a plan proved not to be an effective approach. Some group members suggested we each come up with a depiction of what we believe encapsulates the computer vision in our system, and afterwards we would have a group conference that allowed us to come to a consensus about what we're actually trying to achieve with the face detection/recognition component. In the end, we have opted to code a program that will take images from the cameras as input, feed the data into a face detection recognition machine learning model, and output a value depending on the results of the model.

We also acknowledge that since our security system includes tracking and response values that administrators may need to verify the results of from time to time, there needs to be some effort allocated towards how we plan to interface/communicate the output of the data in a digestible manner such that humans can readily understand. While we have spent time drawing some representations of these applications at this point, our team is on the same page regarding the functions we want the accompanying application to be capable of performing. Since the application serves to mainly display the information to users, our group is confident that the functions will largely mirror some basic applications we've worked on in the past.

Once we were set on the objectives that our code has to meet by the end of this project, the team spent some time reviewing the possible languages we have identified from the research in this documentation to code the computer vision model that performs our classification for the faces in the images fed to it. The Python and C++ object-oriented programming languages seem to be frontrunners for satisfying the goals of our code, and thus we are striving to write our code using either of those languages. While the Java coding language comes in as one of the group's more acquainted languages to work in (due to previous coursework done in that language), research has shown us that extensive open-source computer vision libraries and packages are oftentimes more available and accessible for the Python and C++ languages. Thus, our decision to code in these languages largely stemmed from the existing libraries that these languages are capable of utilizing.

Similarly, the language used by the application for interfacing the data output by the computer vision component in our system has been modestly discussed in previous sections (see 3.2.1.3, 3.2.1.4). From past courses and personal experience, JavaScript has ended up being the preferred language for front-end work on web applications. In a unanimous decision, the group has elected to select JavaScript as our coding language for our web application. Since it is likely that users of our system will defer a few people as administrators to the system, we'll need a database to store information such as accounts and information from the beacon about tracking data of each person involved. The database 'language' typically falls under two categories: SQL or NoSQL. In this project, we'll be using a NoSQL database to store information pertaining to our system.

There is an abundance of applications out there that allow us to write, compile, and debug programs in an effortless manner. Our group largely believes in using a single application to write code for our project in, which ensures we are all seeing the code text and syntax highlights similarly on our own computers. The various classes each of us had to take in previous semesters has shown us several code editor applications, and thus we voiced our experiences in them to each other. After some consideration of all the options we had, Visual Studio Code seemed to be the most suitable code editor, as it provides a wide selection of plugins for us to augment our coding experience as we develop the programs for this project.

In terms of the order of code we aim to work on, it made sense to us that we must code our computer vision component first before moving onto the application of our system. Out of the four members in our group, we have two that are particularly interested in working on computer vision. As such, we've allocated most of the machine learning implementation to two, but the other members are free to suggest changes and help where they feel appropriate. We believe it is proper as the other two members will be working on the hardware side for the most part. The accompanying application side of things will be split evenly amongst the four of us, as it will be the only work left after the core of our system is completed.

For the computer vision code, we have composed the face detection recognition model as an ensemble of a few pre-trained models provided by existing open-source libraries. The idea behind this is to make use of already existing pre-trained models provided by libraries, since it is open source. We believe this is better than trying to completely build our machine learning model from scratch, both that it will save time and ideally make the system more accurate. On top of this, we have created our own model for face classification during the time we work on our project, attaching the resulting model with the ensemble of pre-trained models as well. Encapsulating this final pre-trained model in an object will be beneficial should we need to use the model in other instances (i.e., the microcomputer operation). Once we have the model prepared, our group will proceed to code the microcomputer will be operating on.

The face evaluation program should mainly be running on the microcomputer and needs to operate at all times while the system is in place. To code this, we have been relying on the pre-trained model object instance to perform computations in this code. Outside of that, this portion will need to use libraries that can read images fed by the cameras, and since it is expected to perform calculations for all cameras, will need a data structure that can keep track of information pertaining to the data output by the ensemble model. This code also has to send the images fed to it on the database as we intend to keep recording history of at least 48 hours in the past. Again, our group will utilize the help of libraries/packages to implement this functionality in a straightforward manner. The other information it needs to send is to the application, which should provide updates to users about the tracking data for each user in the system, as well as any warning signals if the computer vision component outputs such signals.

The application code looks to be fairly straightforward to write up after we have the system up and running. The absolutely necessary components include features such as registration of a user, logging in as a user, and displaying information to the user. To do this, we have written the APIs necessary for the website to hit when the user attempts to perform any of those actions. The application will also be communicating with the user database, and as such we'll need to incorporate database credentials in the API we write. The database will consist of the location histories of each member in the system, and a value that changes depending on whether a warning is passed or not by the computer vision component. There will also be a system in place utilizing two-factor authentication to verify users trying to access the private information. Secondary features will be regarding editing/modifying the information displayed, updating the credentials of an existing user, and removing a user from the system.

With tracking the Bluetooth, is it important to display the locational data to monitor indoor tracking through our system. Using our beacons and the tag we are able to obtain the Python Bluetooth library. This library will help us create a socket in which the tag will transmit NMEA format positioning information. Using this standardized format, we are able to manipulate the incoming data to obtain the latitude and longitude of the tag itself every couple of seconds. After successfully obtaining the data points of the tag, it is time to generate the pathfinding design for these positioning coordinates to map to the building room location. This will be implemented if more time is given, for now we primarily focus on proximity tracking and letting us know if the beacon detects a tag in the area, in which the tag will notify the user they are being tracked and sent the approximate location data to the database via UART. To make our indoor tracking come to life and display it we firstly have to create a virtual structure of the building that is being navigated through, which can be done through a JSON format or multiple existing pathfinding tools such as A* in python. Since we are working with only 3 beacons and 2-4 tags, we want to keep the building data small, a small room with a couple of obstacles such as a table and couch for testing. The virtual structure will be stored in the MongoDB database and will be two-dimensional to scale. Once stored we can place our beacon in that JSON format and give a relative location as to where the tags are and display it, although it is not the best approach to go about it, we feel that it is the simplest.

The last thing of concern in terms of the application code is coding the design of the user interface itself. In this regard, we leave this portion for the end since we prioritize having other components fully functional before worrying about the look of things on the website. We have some ideas of how we want to format different screens on the web page, but as for color palettes, animations, and word styles, we plan to incorporate a particular styling that makes our website easily readable regardless of the computer it is being viewed on.

Should a group member lose track of what needs to be done, it is useful to keep a collection of each component that will be worked on and what each component will consist of.

7 PROJECT PROTOTYPE TESTING PLAN

In this section the planning behind testing our prototypes for our hardware and software systems will be discussed. While discussing our subsystems, the testing procedure and the expected testing plans will be examined. This documentation of specified planning will come to life when building our project, providing the team with a better understanding of all the working components. Moreover, any problems or issues that come into light when testing our hardware and software systems will be corrected and documented. The following plans are a broad overview, more information will be added as new testing parameters transpire.

7.1 Hardware Specific Testing

Hardware testing will be done as a team in the senior design lab located in ENG 1 room 456. The senior design lab will provide us with testing equipment such as oscilloscopes, breadboards, and multimeters. These tools will be used to ensure that our hardware components are working as expected prior to putting everything together. Prototype testing must be done on the hardware side to ensure that none of our components that we order come damaged or unresponsive. In the event that parts do not work we are to return them as soon as possible and order the components again and repeat testing steps for them. It is important to note that in the case that we have to return something, we have to estimate the time it would take us to receive the package again, which is why it is important that every part is order prior to the beginning of senior design two so that we have sufficient time to test and build our indoor smart security system.

In this section we go over how our team plans on going about testing out hardware that is essential for this project. Each testing will have their own list of testing materials needed to ensure functionality. We also set a brief outline of the procedures/steps to take when testing to make sure we are doing everything right. Again, these are just a rough overview and procedures to follow when testing, things are subject to change depending on any unforeseen circumstances that come up, as stated before some material that we acquire has the chance to not work therefore we have to most troubleshoot or reorder the hardware material. At the end of each testing, we specified the expected result for each hardware system. We have discussed each individual component testing as well as the combined subsystems to ensure that all the components are able to work together as expected. Below is our plan for our prototype testing for our smart security system.

7.1.1 Camera Testing

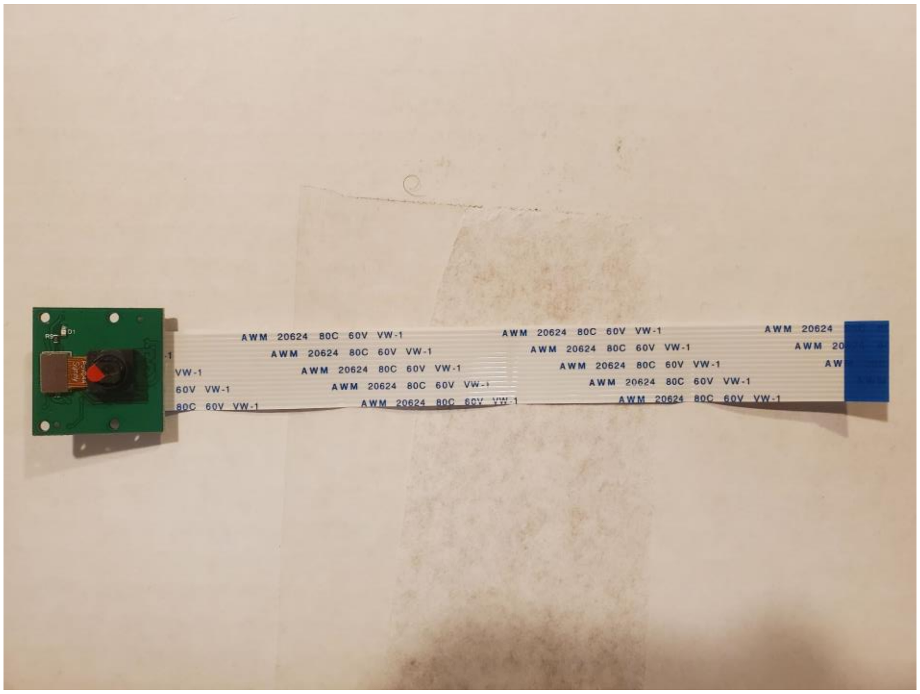


Figure 33: Raspberry Pi Camera

Since the camera is a vital piece of our smart security system, we must ensure that upon arrival, that the camera is working properly.

Test Materials:

- Functioning Laptop/PC
- Raspberry Pi HD Camera

Procedure:

1. Ensure that your Laptop/PC is working
2. Plug in the camera module to the computer
3. Install camera raspberry pi software
4. Run the camera software - adjust settings if needed

Results:

Camera is able to take pictures and videos in real time.

7.1.2 Servo Testing

The servos are responsible for moving the camera to provide a better view of targets within the field of vision. Upon arrival, testing the servo platforms is essential.

Test Materials:

- Functioning Laptop/PC
- Arducam Pan Tilt Platform

Procedure:

1. Ensure that Laptop/PC is working
2. Plus, in Servo module to the computer
3. Plug in the camera module to the computer
4. Install Servo testing software
5. Run the Servo Software - adjust settings if needed

Results:

The servo is able to move the camera platform left and right. The servo is also able to tilt the camera forward and backwards without any issues.

7.1.3 Antenna (Beacon) Testing

The beacons will send and receive data at a set rate (in seconds). This testing will be performed along with microcontroller and microprocessor testing.

Test Materials:

- Ammeter
- Voltmeter
- Oscilloscope
- CR 2032 Battery
- 3x Beacon

Test Procedure:

1. Ensure Oscilloscope is properly grounded
2. Test battery to verify voltage capacity
3. Measure voltage and current outputs of PCB
4. Place battery into slot to power the PCB
5. Measure voltage input and output
6. Verify that the chip is transmitting a signal and record the frequency

Results:

After testing the beacons, we can transmit a signal sending a code that will be the device's UUID which will be used in later testing. Once both the beacon and the microcontroller can work together, then we need to combine these efforts with those of the microprocessor controlling our camera system.

7.1.4 Microcontroller Testing

The microcontroller will be responsible to communicate via devices via UART to transmit data. This testing to ensure that our microcontroller is working as expected without any issues.

Test Materials:

- Ammeter
- Voltmeter
- Oscilloscope
- Breadboard
- Microcontroller
- Function Generator or Laptop/PC

Test Procedure:

1. Ensure that the Oscilloscope is properly grounded
2. Connect microcontroller to peripherals (Oscilloscope, breadboard, voltmeter, and ammeter)
3. Test different voltage input/outputs and measure frequencies
4. Using testing software, plug the microcontroller in and test
5. Ensure that beacon is receiving a signal from the microcontroller

Results:

After running the tests on the microcontroller, the functionality for the sender/receiver component of the project should be complete. Based on the location in relation to the microcontroller, our application should show the indoor location of each beacon within one meter of accuracy within the received RSSI.

7.1.5 Microcomputer Testing

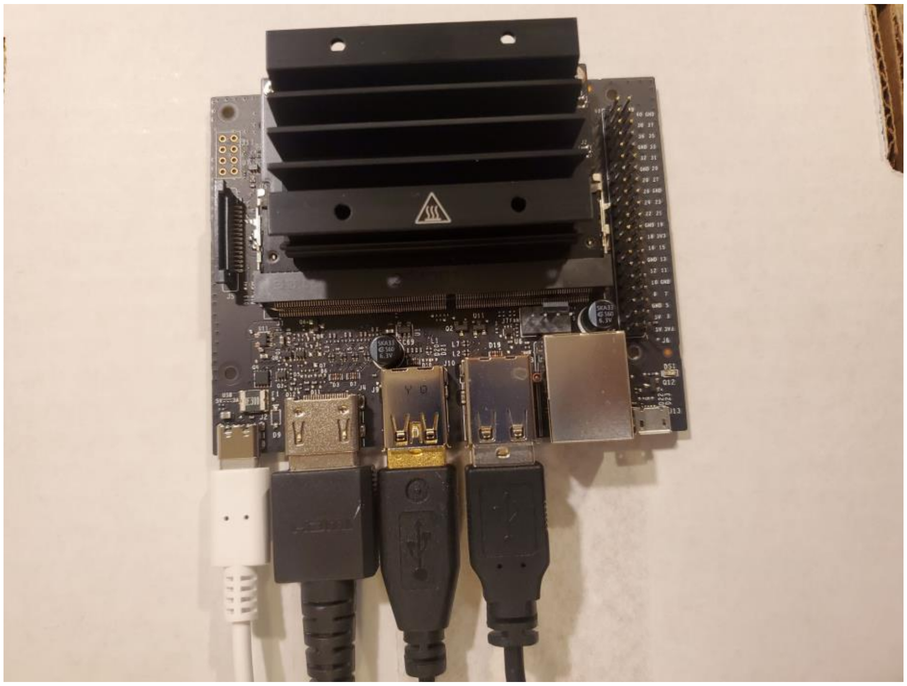


Figure 34: Nvidia Jetson Nano

Test Materials:

- Peripherals: Mouse, Keyboard, Monitor
- Servo Motors (To test PWM capabilities
- Deep Learning Inference Benchmark (To test speeds of device)
- M-F / M-M / F-F connectors (for connecting servos to pins)

Procedure:

1. Setup Jetson Nano OS/Test environment as stated by NVidia
2. Ensure that peripherals work properly
3. Test serial capabilities with simple servo motor communication

4. Test processing speed using deep learning benchmarks to ensure device is functional under high load

Results:

After taking the Jetson Nano out and setting it up, testing all peripheral functionality is important. After testing the interfacing tools, then we need to verify that communication between the microcomputer and the servos is happening the way we expect it to. Next, comes developing and testing suitable algorithms for the deep learning component of the product. After several test runs of the algorithm that runs the fastest and most efficient. Our second test ended up failing which led us to an alternative route for our facial recognition and face detection system.

7.1.6 BLE Tag Testing

Test Materials:

- BLE Corresponding Tag
- BLE Corresponding Beacon

Procedure:

1. Setup and complete prototype testing steps for the BLE Beacon
2. Setup and follow instruction for the BLE tag setup
3. Check the Tag LED when coming in proximity of the BLE Beacon

4. Check the Tag LED when coming out of the proximity of the BLE Beacon

Results:

After BLE beacon setup, the beacon should be able to emit Bluetooth signals, which can be picked up by any device with Bluetooth capabilities within the area. After the BLE beacon setup, the Bluetooth corresponding tag should be able to detect and receive the Bluetooth signals. The tag we purchased has visible LEDs which shows when receiving Bluetooth signals. If the LED shines green when coming in proximity that means that it successfully receives signals, if the LED shines red when coming out of proximity that means that it's not detecting any signal, which is accurate when moving away from the beacon.

7.1.7 Beacon - Tag system Testing

The beacon-tag system is responsible for the indoor tracking implementation and is a major subsystem in our project. This system has three beacons and two tags, one being a corresponding tag with the beacon and the other being the microcontroller with the embedded Bluetooth module.

Test Materials:

- 3x Beacon
- Beacon corresponding Tag
- Microcontroller (ESP32)
- CR 2032 Battery
- 3x Beacon

Test Procedure:

1. Ensure Beacons are in are with nothing in the way of signal emission
2. Test corresponding tag to see if signal is being received
3. Place battery into slot to power the PCB for microcontroller

4. Test microcontroller to ensure it getting Bluetooth signal as well
5. Verify that the chip is transmitting a signal and record to our IoT

Results:

The tag is able to receive the emitted signal with no issues and it will indicate to us with its LEDs incorporated on the tag. Similarly, the ESP32 has LED to display Bluetooth connectivity, if working properly the ESP32 should be powered on by the PCB LiPo Battery and should be able to receive these signals and it will be indicated to us by the onboard LED which will flash blue when connecting and no flash when not detecting any signal.

7.1.8 Camera - Servo Testing

The camera - servo system is responsible for mounting and housing our raspberry pi camera. The servo is responsible for moving the camera around into positions to acquire a better image for our facial recognition software.

Test Materials:

- Arducam Servo
- Raspberry Pi HD Camera
- Nvidia Jetson Nano

Test Procedure:

1. Ensure Nvidia Jetson Nano is plugged in
2. Connect the Servo via the control board provided
3. Screw in the camera into the servo platform 5-8mm screws
4. Connect camera via connection ribbon to NVidia Jetson Nano
5. Connect the Nvidia jetson to laptop via USB to upload code samples from Arducam
6. Upload base code from Arducam to test servo functionality
7. Using laptop test Camera field of view and adjust if necessary

Results:

The servo is able to successfully carry and hold the raspberry pi camera in a still position prior to servo movement. The servo should be able to move limitlessly without losing any torque or rotation freedom once the camera has been screwed into platform position. The servo should be able to be manipulated by the jetson nano once the code has been uploaded to it. Once connected to the jetson, the camera should be able to display its field of view once the USB connection has been established with the jetson and a working laptop.

7.1.9 Battery Testing

Implementing batteries into the project saves time, space, and cost looking at the bigger picture. With the “coin batteries”, the beacons in our security system should last between one to two years before needing replacement. Any investor would benefit from this reduction in recurring cost. The batteries also allow devices to receive power without needing to take up more space than the electronic device needs to operate to its potential.

Test Materials:

- Multimeter
- 3.7V Lipo Battery
- Battery Pack (Will use these if multiple MCUs present)
- PCB
- MCU (If Applicable)

Procedure:

1. Ensure that the multimeter is grounded
2. Test voltage on LiPo battery to make sure that it is not < 3.0v
3. Test voltage on battery pack to make sure that it is not < x.0v
4. Once complete with step 2, place battery in beacon
5. Once complete with step 3, connect battery pack to power module on MCU

Results:

Batteries play a large role in the success in the operation of this project. Having the ability to power wireless devices is crucial for the design component. Beacons will need to be small enough to be undetected and be able to remain in that place without constant maintenance (both of which are achieved with battery powering the beacon).

7.1.10 BLE Signal Interference Testing

Getting accurate RSSI data is crucial to our project as it is the main component for our trilateration algorithm. It is important to select the correct variable ‘N’ for our Distance formula. ‘N’ corresponds to the number of environmental factors in our area. So, we ran a test to determine the optimal value for ‘N’.

Test Materials:

- Second Gen PCB design (Beacon)

- Second Gen PCB design (Tag)

Procedure:

1. Place Beacon in stationary position via USB-C connection
2. Place moveable tag 3m away from beacon
3. Track distance formula results with incoming RSSI with no interference between.
4. Track distance formula results with incoming RSSI with human interference between.
5. Track distance formula results with incoming RSSI with wall interference between.

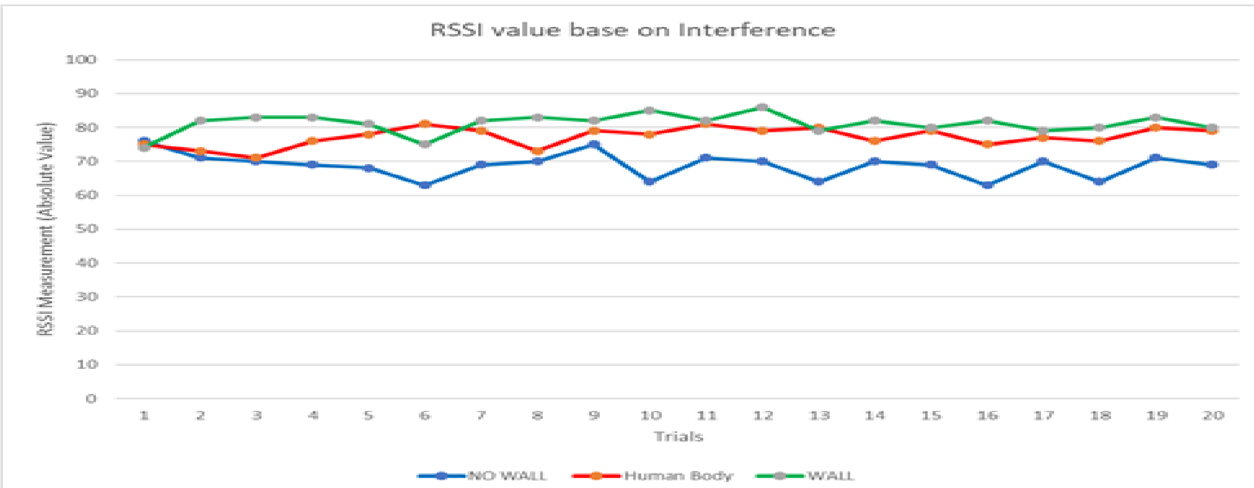


Figure 38: RSSI Value base on Interference

Results:

The blue line represents the signal with no physical interference between the beacon and the tag, the Orange represents human body interference, and the green represents wall interference. The blue line represents the strongest signal strength at an average of -68.8 for the RSSI. With Bluetooth signals the value of RSSI can fluctuate when there is a physical object between a transmitter and a receiver. When testing the effects of a human body or wall between the tags and beacons we observed an average decrease of 8.45 and 12.35 respectively. In situations where a wall or human body will be between the tags and beacons, we can adjust for the environmental factors in our distance equation. For this implementation we decided to select a value of 3 for our environmental factor variable 'N'. These tags will ideally be carried in the pockets or close to the body of the monitored personnel, so to calculate accurate distances we must consider that the incoming received signal strength is inflated due the environment around the beacon and the tag.

7.1.11 Indoor Positioning Accuracy Testing

Once we got the estimated 'N' calculation, we are now ready to test our Indoor Positioning System. We had to see how accurate our distance formula was at estimating how far the tag was from the beacon, so we ran the following series of test.

Test Materials:

- Second Gen PCB design (Beacon)
- Second Gen PCB design (Tag)

Procedure:

1. Place Beacon in stationary position via USB-C connection
2. Place moveable tag 10m away from beacon
3. Track distance formula results with incoming RSSI

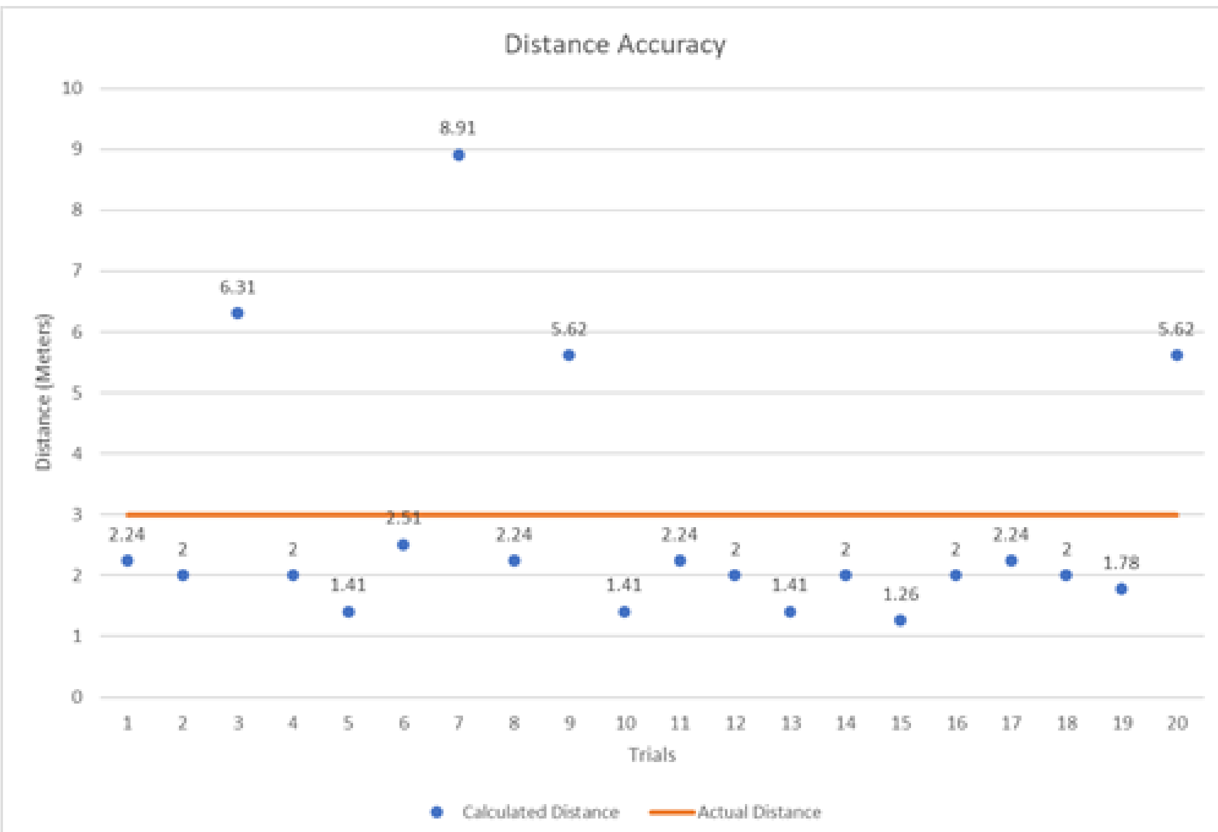


Figure 39: Distance Accuracy test for Beacon

Results:

Figure 39 represents the results of the testing to check that the variance for the calculated distance was acceptable. In this test we set a tag 3 meters away from a beacon and recorded 20 calculated distance values. When averaging out the calculated distance over this span we found that on average the system calculated a distance of 2.86 meters (14 cm error) which is within our engineering specifications for the positioning system.

7.2 Software Specific Testing

When it comes to preparing our software for use with tasks necessary in our project, such as face detection, face recognition, and object detection, some instructions to get everything up and running were followed. Since we're using a microcomputer to perform the essential operation

7.2.1 Database Connection Testing

Having a location to store all the images and user information is essential when implementing object recognition. In our case, a database is needed to store images and information of individuals so that when we connect our facial recognition software to our database, it can pull the information from the database.

Test Materials:

- Functioning Laptop/PC
- Server database (MongoDB)
- Database managing software (JavaScript)

Procedure:

1. Ensure Laptop/PC is fully functional
2. From the database server, create a Table
3. From database server, add test data/images
4. Run the JavaScript program to create connect to the server
5. Test connection and display the testing information from database using the JavaScript code

Results:

The JavaScript is able to connect to the server and pull out the information within the database. If the code is unable to connect to the database, it will display an error.

7.2.2 Mobile and Web application Test

A primary feature we wish to incorporate with our system is web and mobile applications where all content and functions are free to access with a created account.

Test Materials:

- Functioning Laptop/PC
- Website

- Smartphone
- Mobile app

Procedure:

1. Ensure Laptop/PC is fully functional
2. Access site from Laptop
3. Ensure site looks like the following:
4. Ensure Smartphone is fully functional
5. Download and open App from app store
6. Open app and ensure it looks like the following:

Results:

Webpage should appear as designed and fit to scale with bootstrap. All features should be accessible by the user and fully functional. Mobile should appear as design and fit to scale with bootstrap as well. All features must also be fully accessible by the user. App must be able to transition from horizontal to vertical display.

7.2.3 Web Application CRUD Testing

It is important to develop an application where users will be able to upload images of individuals they want stored in the database. We want users to be able to create records of images, receive information based on their account (Admin/employee), update any personal information, and be able to delete any database input completely if it corresponds to their account.

Test Materials:

- Functioning Laptop/PC
- Server database (MongoDB)
- Existing web app
- Existing profile

Procedure:

1. Ensure Laptop/PC is fully functional
2. Access the web app with following link:
3. Login to the site
4. Upload image to database using application
5. Identify user information and be able to update
6. Delete any image entry that existed prior to session

Results:

The web site is able to connect to the server and pull out the information within the database. If the site is unable to connect to the database/server, it will display an error. If connected, the user should be able to sign in and update their profiles with images, and personal information. Users must be able to confirm that any update or creation of data input has successfully registered to the database without any errors. For deletion, the select data part must be completely gone from the user's profile.

7.2.4 Computer Vision Program Testing

The following test is designed to test our computer vision program. Using our camera, once we start running our computer vision program, the code should proceed to read input from the cameras and perform the face detection & recognition.

There were two metrics that we investigated to understand the proper conditions for the camera to operate in:

1. Effective Distance Range (at 800 lumens)
2. Effective Light Range (at 7 ft)

Here is how we prepared to conduct the tests:

Test Materials:

- Functioning Laptop/PC
- Camera Module
- Python IDE

Setup Procedure:

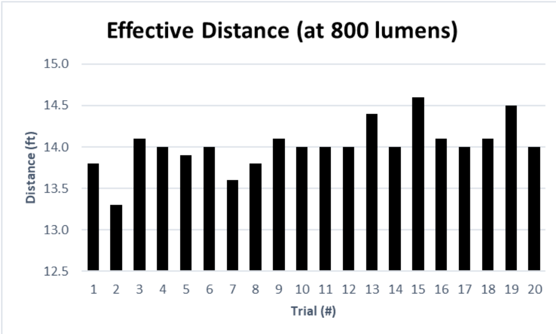
1. Ensure Laptop/PC is fully functional
2. Open Computer Vision program in preferred IDE, compile the code.
3. Run the Python program
4. Observe the new window that pops up showing the frames captured by the camera and the results of computer vision reflected.

Effective Distance Range Procedure:

1. Have one person in the camera walk slowly away from the camera.
2. Record the distance at which the camera stops being able to locate/recognize the person.
3. Repeat for 20 trials.

Effective Light Range Procedure:

1. Have one person in the camera stand 7 ft from the camera.
2. Begin lowering the brightness of the lights in the room.
3. Record the brightness at which the camera stops being able to locate/recognize the person.
4. Repeat for 20 trials.



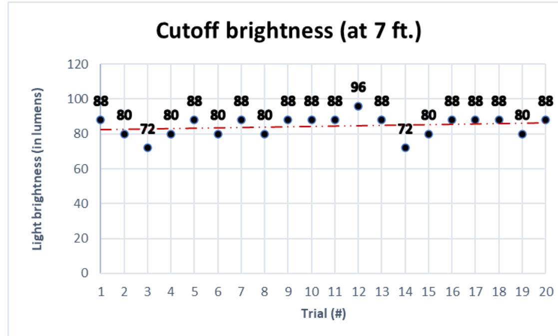


Figure 40

Figure 41

Results:

Figure 40 depicts several trials of the effective distance, where the lights used are kept at 800 lumens (100% brightness). Note that on average, we see the effective distance at this brightness is 14 ft. Trials that performed slightly better/worse than average can be attributed to factors such as blurriness due to auto-focus, sunlight making the room brighter, and different locations for testing.

Figure 41 depicts the results of the light range trials were conducted using Philips Hue A21 light bulbs, and by varying the brightness, this has shown that around 80-90 lumens is when the computer vision will fail on a face. Some outliers can be attributed to the blurriness of the camera at that point, or sunlight increasing the overall brightness in the room.

8 TEAM RELATED

This chapter will discuss the team related materials including an overview of our administrative content, and a write up on each personnel working on this said project. This chapter will go over our future plans as well as specific individual motivation for why we selected to do this project.

8.1 Administrative Content

In this section we discuss various different topics related to project management and personnel biography of each individual team member. We discussed the major milestones for this project, go over our Gantt chart, and discuss the budgeting for this project. All of these major components were responsible for managing the project and keeping the lens within the scope of our project. We encourage similar discussions for future groups working on a Senior Design project of any kind, as it is important to consider in any developmental process.

8.1.1 Milestone discussion

Milestones are great to have for any major project as it not only displays what the team manages to accomplish, but also sets a strict timeline that we must follow to motivate the team to complete designated tasks on time and within the budget. We wanted to build our milestone to be pretty flexible so that if needed we can modify our milestone if any issues arise. We plan on taking an agile approach to this project to not only be more efficient, but to prepare each team member with an industry standard on how to approach projects. Our milestone figure was displayed in previous sections above, and again we plan on sticking to it as those cover the important due date and upcoming task, we have to complete by the end of senior design two. As we continue to move forward and more components come in, we have made changes to our charts and milestones to implement specific due dates for when we want parts of our projects done.

8.1.2 Gantt Chart

A Gantt Chart is essential for our project. A Gantt chart is a project management tool that helps the team delegate tasks and stay up to date on the member's and task progress. Below you will see our project timeline and deadlines for the task that we wish to get down and the time we would wish to have it done.

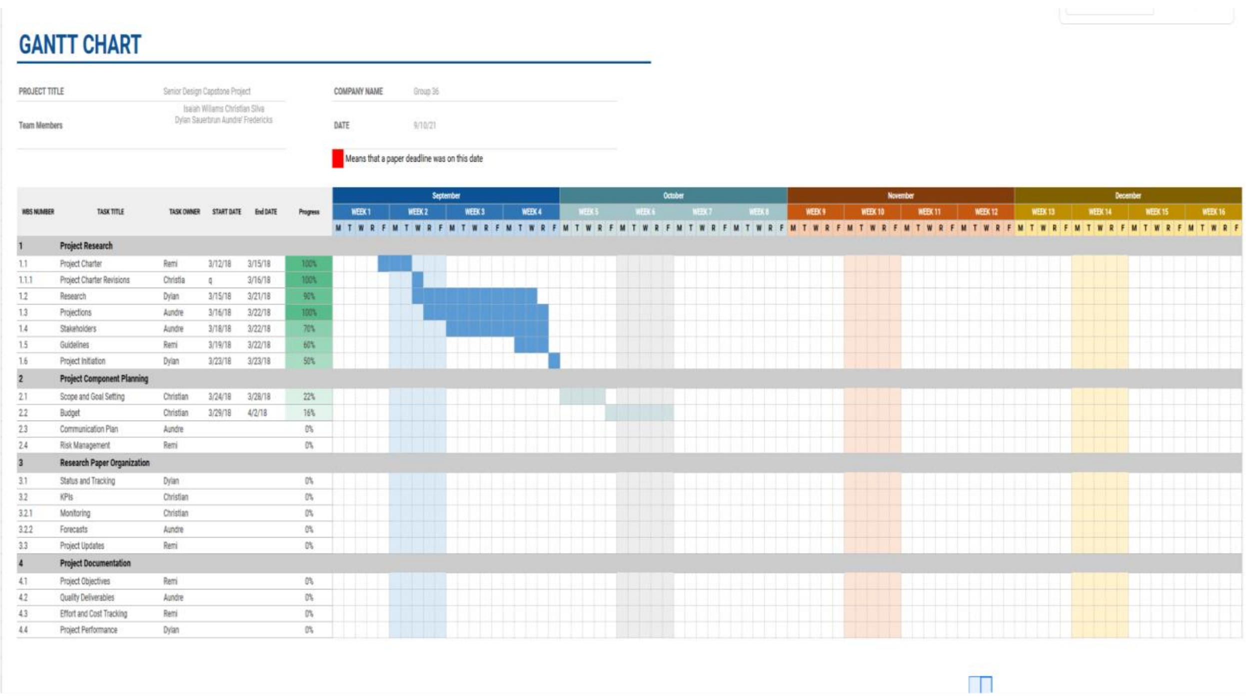


Figure 35: Gantt Chart

8.1.3 Group Communication

Within any functioning group we believed one of the strongest values to have been a strong sense of communication. This manifested in our group by having two hour-long meetings twice a week to keep everyone informed on the progress each person has made. The goal of some meetings was to either discuss particular aspects of the project that needed to be worked on and/or clarified, and other meetings were spent researching together, relaying possible pieces worth considering to those who may need it.

The benefit of having a strong connection between all of our members displayed in scenarios that would have otherwise been a hassle to handle; for example, if one member were to become sick and could not make a meeting, the other group mates would make it an effort to document anything discussed in the following meeting. Other times when unforeseen circumstances would happen to a particular member was also aided by this approach.

Our primary form of discussion came from the online communication service known as discord. In discord we made tons of channels to collaborate, these channels include research, group links, agendas, meeting plans and a general channel. Discord is where we held our meetings and communicated important ideas to one another. We are also all exchanged phone numbers in case of emergency, we all know how to reach each other. In the beginning we laid a general foundation and ruleset for one another to make sure everyone stayed on task and participated in collaboration, since each person is responsible for specific parts of research. Some of the rules include the importance of response time. We asked each group member to ping each other and give the member around 2 - 3 hours to respond before you text their personal number. We also set a rule that if more than 4 meetings were missed by a team member that the group will have to make a choice on whether or not to report the member to Professor Ritchie. The most important rule was to please collaborate and express your ideas. Share what you have researched and make sure you are writing about what you said you are writing about, basically meaning to make sure you do the work you said you were going to do so that the group does not get blindsided and know what is yet to be done.

8.1.4 Budget and Finance Discussion

Our team decided to fund our own project because we believe that we can keep the price range pretty low at around $350 and below, plus half of our project turns out to be software which for the most part is free. Our total cost would have been more if our team had to purchase testing material, but thanks to the University of Central Florida and the College of Engineering and Computer Science they provide us with a well-equipped lab filled with tons or equipment. Our professors are also kind enough to lend our team extra testing materials known as the discovery kit, which comes with circuit components such as wires, resistors and capacitors that we can use for our testing. The discovery kit also comes with extra breadboards and a multimeter to read voltages.

Our team consists of four members, meaning that the total cost will be split evenly across all four of us. For the parts that we acquire and use for these projects, we plan on donating it to the upcoming senior class since no one in our group will have future use for it. We want to encourage upcoming classes to take an interest in Bluetooth technology and learn that it is not an imitating technology. By donating our materials, we hope that a future senior design team can follow in our footsteps and do a similar approach to indoor tracking, perhaps a different way from our implementation to show how flexible and free Bluetooth technology really is. Our team's main goal is to experience what it feels like to become a team of engineers and work together to solve a problem we came up with and implement our solution during this senior design one and senior design two classes.

It is important for our team to remember within the budget and time constraint for this project. Saying between these boundaries will help our team learn a sense of industry balance and time management which is essential when we go out into the real world and become professional engineers. Some group members have had experience with internship and research, so they know how important finance is. Companies or research always set boundaries on how much you can spend for a certain solution because they do not want to overspend or be taken advantage of. In future opportunities and interviews, this senior design project becomes a huge point of discussion for all of us, as it expresses how we are able to work together and collaborate to solve a problem, and also shows how we were realistic about this and set budgets and time constraints for ourselves to deliver this solution.

We know that our budget cannot account for everything and anything, as there will be, as always, some miscellaneous parts that will be needed, which is why we are providing ourselves with a little bit extra money to cover some of these costs. We believe that in terms of this scope of project that our budget is fair and more than enough to allow our team to get a solution for our smart security system. Again, we are doing this at a whole scale because if we were to develop a solution for a whole building it would be probably around eight times more our cost for all the material, it would also take us a lot more time to complete. As our design comes to a completion and we are near the end of our milestone. We have reviewed our budget and finances and discuss how we did overall. We have seen cheaper and more expensive solutions for indoor tracking in the past and we are confident that we set an appropriate amount for ourselves.

8.2 Personnel

Smart security system consists of four computer engineering students, each of which brings a unique skillset to the project. In this section you will read a brief bibliography of each member of the team who worked on the project and their responsibilities. While all members of the group plan on working on each individual part to assist, each group member has a direct impact in the following areas. Each member was responsible for getting their assigned task done on time and to document any research or testing they did with their component.

8.2.1 Christian M. Silva

Christian M. Silva is a senior computer engineering student who has experience in full stack development. Christian is interning with Leidos as a Software Specialist where he assists his team with front end development, leading to him taking over the web and mobile application aspect of this project. Besides this Christian is also taking Process of object-oriented software development, which is a class dedicated to full stack development which provides an educational background for applications that we plan to build. Christian also plans on helping Dylan alongside the facial recognition software as he is intrigued by the software system and wants to learn more about computer vision.

8.2.2 Dylan Sauerbrun

Dylan Sauerbrun is working on finishing his Bachelor of Science in Computer Engineering at UCF. Throughout his college journey, he has spent most of his time working on projects in class. With an affinity for software development and mobile design, Dylan has put his efforts into different projects using a handful of stacks. In one project, Dylan was in charge of writing the frontend portion to enable communication with the backend. As for mobile design, Dylan has primarily worked using Flutter software to implement mobile apps but is currently seeking to understand mobile development using React.js library in the JavaScript language. Being a computer engineer, Dylan has experienced an adequate amount of material regarding electronics, and as a result he has been searching for ways to get more involved in that field as well. After college, Dylan hopes to start working at a company that can teach him the ropes and provide a clearer idea of what it means to work in an engineering career.

8.2.3 Aundre

Aundre’ A. Fredericks is currently a Senior studying Computer Engineering. He has interned with MITRE Corporation doing full stack web development and has spent two summers interning with Texas Instruments. While interning, he developed automation scripts using python. Aundre will be joining Texas Instrument’s Engineering Rotation program this upcoming summer with a start date of June 2022. He anticipates that gaining experience interacting with the hardware components of the project will help him develop some skills necessary for him to hit the ground running at Texas Instruments. He will also play a role in implementing some components of the full stack as most of his technical experience is in application development. Aundre’s involvement with the computer vision and the implementation of the servos is unknown at this time.

8.2.4 Remi

Isaiah “Remi” Williams is currently a Computer Engineering major at UCF. He has 2 years of experience working under Dr. Murat Yuksel in the UCF Networking and Wireless Systems Laboratory as an undergraduate researcher. He has an interest in the fields of embedded and software engineering with a focus on communications. He has experience working with microcomputers such as the Raspberry pi models 3 and 4, as well as Arduino boards. Remi plans to work on establishing the IoT connectivity for the security system to ensure that sensor data can properly processed by our security system and will also assist Aundre in establishing the beacon functionalities. He will also be designing the PCB power supply for the group.

9 PROJECT SUMMARY AND CONCLUSIONS

In this final chapter we give a project summary and go over our final conclusions for our project. We discuss logistics, our overall goal, and our plan for the smart security system implementation.

9.1 Project Summary

The goal of this project is to create essentially a smart security system. Our smart security system consists of two main subsystems - a beacon and tag system and a camera system that analyzes live feed when people walk within the field of view with our facial recognition software. After coming up with essentially what our project is, our team got together and started to discuss the logistics of our design and see how we can bring this more to life. Our smart security system aims to increase safety and promote productivity in the workspaces for customers that choose our solution. We do not plan on putting on product for sale at the end of this 2-semester course, however all of our findings will be open source to encourage future groups to expand on Bluetooth technology.

The primary goal for this project is to build a functioning prototype of a smart security system that utilizes facial recognition software to detect individuals within building locations and to design an accompanying Bluetooth tag/beacon set to assist with indoor location tracking of users through high secure areas within the building that not everyone should have access to. Our first main system is the Camera - Servo system which is responsible for hosting our facial recognition software. The Camera - Servo system consists of three main components, the Raspberry Pi Camera 12 MP, the Arducam Pan Tilt platform and the Nvidia Jetson Nano (2GB). These three components all work hand in hand, the servo is responsible for mounting and hosting the camera and moving it around 180 degrees and tilting it forward and backwards as needed. The Nvidia Jetson is used to program both the servo motors and the camera and also responsible for storing anything we find valuable to memory during the time of usage. The camera is to be connected directly into the Jetson Nano via the CSI ribbon input. The camera-servo system main functionality is to put the camera in the best position possible to recognize.

Our secondary system is the Beacon - Tag system which is responsible for the indoor tracking portion of this project. The beacon and tag we purchased both go hand in hand, using the same Bluetooth protocol and version 5.0. Our team wanted to use the latest Bluetooth version but due to the lack of research with indoor tracking behind it we decided to leave it and do Bluetooth 5.0 instead when it still has relatively the same features as Bluetooth 5.2. From our understanding, we can update these beacons and tags to 5.1 but not the latest 5.2, which is probably what our team will end up doing after various discussions. Our end tag will end up being the ESP32 microcontroller due to its built in Bluetooth, it should be able to detect the emitted signals from the three Dialog DA14531 BLE Beacons. Once the signal is detected, we plan on encoding the microcontroller to send us data via UART to the cloud/database to store the information so that we can populate the database table in our web application as soon as in the previous sections above. Overall, the connection between the microcontroller, beacon and tag is vital for communication to acquire the indoor location and to accurately display that information on our site. For the moment we plan on just having a table that shows which tag number has been where and the timestamp of the most recent emission signal. In the future, if time allows us, we do want to implement some feature to bring our live indoor location to life which includes building a room layout and mapping out exactly where the tag has been in relation with the beacon. To fulfill our PCB requirement, we are designing a PCB to power on the microcontroller - the ESP32 chip. The PCB will be basic and will be designed by our team, if we run into any issue our point of contact is Professor Samuel Ritchie. Our PCB schematic and plan can be seen in the previous sections above.

Our team also plans on creating an accompanying web and mobile application. The web application will have features that show a table of which tags have been detected and their time stamps, the data table will constantly be updated whenever a tag is detected by the beacon and its relative location to that corresponding beacon. The web application is meant to be used by admin, so the admins will have the privileges to add and remove tags within their organization. The admins also have access to upload images to our facial software, so let’s say that the admin wants to add in a new user who is an employee they can add in the image file which will directly go into the training database, which then will process the image to train the facial recognition software system. The mobile application will have similar features to the web app, they are meant to go hand in hand with each other and be able to use and access from any way given that they have the proper credentials to login. We plan on incorporating extra security measures with our site, implementing JWT token for authentication is the plan, along with using SendGrid for email verification and password recovery.

Ultimately, this is a high-level overview of our project that goes into detail with each subsystem. We hope that we are able to replicate what we put on paper and design our project with little to no bumps on the road while building our functional prototype.

9.2 Project Conclusions

As stated, before our primary motivation for this project was to encourage every team member to step outside their comfort zone and work on a project that tested them in a field that they have had no prior exposure to. This was a motivation because for the team, this project will be a main focal point of discussion in future interviews and job applications, so we want to make sure we are able to work as a team, with a problem we are not familiar with and come up with our best possible solution - and that is what we feel we have accomplished thus far in our project. Our team is happy with our selected project as it is in a sense an interdisciplinary project done by 4 computer engineering students. It's something that all find interest in working in which is good because, if anyone in the team is not interested then we have not gotten one hundred percent out of them that we need for senior design 1 and senior design 2 completion. Again, a primary motivation for this project was to fill our resumes and explore technologies that are used in today's technologically advanced world. Originally our team found a couple of roadblocks with the implementation of indoor tracking and facial recognition. Originally, we wanted to use RFID, but after discussing this with our advisor, we were told to look into Bluetooth as RFID is too simplistic of a technology and will not give us the solution we are looking for. Also, in the beginning there was a big misunderstanding between facial detection and facial recognition, we decided to stick with facial recognition because we felt that it made more sense in terms of security measures to take for this project.

The primary motivation as mentioned before is to go into a field that our team has not had much exposure to, to show ourselves that we are able to pick up and learn new technology when deemed necessary. Which is huge for our future as we become professionals. Everyone in the group is well aware that the industry of engineering is always advancing with new technologies, so it is important as individuals that we learn how to adapt and learn new technology that we are not familiar with. Also, this project will be a huge focal point of discussion for our future interviews as it is our first time ever. We are a big scale engineering project using everything we learned from prior classes. Overall, this project serves as a milestone for us all as it connects our academic career with our future professional career as engineers as we sit to learn and test software and hardware systems while maintaining effective teamwork.

The purpose of this project was to automate the monitoring of secured locations using an Indoor Positioning System (IPS) in conjunction with facial recognition software. This provides a method of tracking an individual's path throughout these locations, while also obtaining the individual’s identity. The IPS component was implemented using BLE 5.0 tags and beacons. These tags/beacons make use of the ESP32 Bluetooth antenna module. With three stationary beacons and at least one tag, trilateration can be used to approximate the position of the tag using the beacon's RSSI values. The computer vision component uses Python, OpenCV and the face recognition library to detect and recognize faces of people that enter a location. This identification info is then sent to our software application. In this software application alerts are created using the data obtained from our edge devices (camera, beacons, etc.), and all of this is done in real-time to provide the most accurate information about people’s whereabouts.

Our system was able to determine the position of an individual with approximately 1 meter of accuracy. And when interference is expected, we can account for this by altering our RSSI calculations accordingly to calculate the values more accurately. Although we can account for this, the security system is best used if there are no solid walls between the Beacons and Tags to minimize signal loss. In regard to the computer vision component, the biggest environmental factor was the brightness of the room, and testing found that ~80 Lumens was the minimum light level at a distance of about 7ft, and at 800 lumens the maximum detectable range was about 14ft. The alert functionality was important for this system as it lets the person monitoring the software know when there are security infractions being committed. The system is able to detect if users are cleared for access based on their face and/or tag. The system is also able to detect if users are getting too far from or too close to the secure location as well. And the facial recognition system acts as a last point of failure, so if a person is caught by the camera trying to enter the area without a BLE tag being detected, an alert will be sent to the application.

The system was successful at providing a variety of security alerts in real time, but there is still room for improvement in the development of a system like this. Getting more accurate RSSI is crucial to obtaining more precise position data, and one way of getting that RSSI is to use stronger Bluetooth antennas. Another alternative communication protocol is Ultra-Wide Band, which provides low energy usage while also detecting small changes in distance and direction. Another improvement to consider is the use of a camera system with pan-tilt-zoom functionality as it provides much more coverage and recognition range than the current system’s implementation.

In conclusion, our team aims to fill in gaps of our professional careers while actively working on a project we are interested in. Our project once completed we plan on making it available to use and repeat given permission from any of the team members. We encourage future students to take an interest in Bluetooth indoor tracking and to take a step out of their comfort zone and work with technology they are not familiar with as an invitation to a new challenge. If any future student has an interest in this topic, feel free to email any of the team members for the source coding and testing for our items. We hope that future class and we included gain valuable knowledge in the system engineering field. Our team is ecstatic to go from the researching phase to the designing and building of our functioning prototype.

We would like to thank professor Samuel Ritchie, and Professor Lei Wei for comments and opinions about our projects. We would also like to thank our family and friends who have supported us through our academic careers and keep pushing us further and further to reach new goals. Their commitment and dedication were a key factor in motivating us to strive for success for our project.

APPENDICES

Appendix A - Copyright Permissions

SKU: B0191PT

Christian Silva <chris1122.cs@gmail.com> Thu, Dec 2, 2:00 PM (4 days ago)
to support

Good Afternoon,
I am Christian Silva, a Computer Engineering student at the University of Central Florida. I am currently in senior design and researching facial recognition systems for our project. We plan on using servos, a Jetson Nano and Raspberry Pi camera module. We came across your document for the Pan-tilt kit that we plan on purchasing. My team and I wanted to get authorization to use the images and data table seen in the user guide document for this product.

Please let me know if you have any questions or concerns

Re: [Non-Order] Christian Silva chris1122.cs@gmail.com Inbox

Irene Fri, Dec 3, 3:39 AM (3 days ago)
Hello Christian Silva, Thanks for contacting Arducam. Please let us know what we can do for you? Thank you. Best regards, Irene From: Arducam <info@arducam.com>

Christian Silva <chris1122.cs@gmail.com> Fri, Dec 3, 8:21 AM (3 days ago)
to Irene

Hey Irene!
Thank you for getting back to me! I am currently a senior design student at the University of Central Florida working on a research project. We plan on using the pan tilt servo platforms you guys have! We would like to cite your page and use some of the images in the documentation for our technical report and would like to request authorization to do so.

Please let me know if you have any questions or concerns!

Best,
Christian Silva

Jetson Nano schematic

Christian Silva <chris1122.cs@gmail.com> 8:54 PM (0 minutes ago)
to support

I am currently a senior design student at the University of Central Florida working on a research project. We plan on using the Nvidia Jetson nano microcomputer! We would like to cite your page and use some of the images in the documentation for our technical report and would like to request authorization to do so.

Please let me know if you have any questions or concerns!

Appendix B - References

[1] Camarato, Steffen,et al. “Autonomous NERF Turret with Facial Recognition” University of Central Florida,2019

[2] Compagno, Andre, et al. “Indoor Navigation System with Beacons” University of Central Florida,2015

[3] The Blavatnik School of Computer Science, Tel-Aviv University, [Online]. Available: https://www.cs.tau.ac.il/~wolf/ytfaces/ [Accessed November 2021].

[4] OpenFace, [Online] Available: https://cmusatyalab.github.io/openface/ [Accessed November 2021].

[5] Arducam, [Online] Available: https://www.arducam.com/docs/cameras-for-raspberry-pi/native-raspberry-pi-cameras/8mp-imx219-motorized-focus-camera-module/ [Accessed November 2021].

[6] Arducam, [Online] Available: https://www.uctronics.com/download/Amazon/B0227PT-Color.pdf [Accessed November 2021].

[7] Jetson Nano User Guide, [Online] Available: https://developer.nvidia.com/embedded/learn/jetson-nano-2gb-devkit-user-guide [Accessed November 2021].

[8] DeepFace, [Online] Available: https://pypi.org/project/deepface/ [Accessed November 2021]

[9] Evolution of Indoor Positioning Technologies: A Survey, [Online] Available: https://www.hindawi.com/journals/js/2017/2630413/ [Accessed November 2021]

[10] Indoor Positioning Technologies, [Online] Available: https://www.leverege.com/iot-ebook/rfid [Accessed November 2021]

[11] 5 Things You Need to Know About Beacon Technology, [Online] Available: https://www.wordstream.com/blog/ws/2018/10/04/beacon-technology [Accessed November 2021]

[12] Bluetooth Location Tracking & Positioning [Online] Available: https://www.inpixon.com/technology/standards/bluetooth-low-energy [Accessed November 2021]

[13] Indoor Positioning, Tracking and Indoor Navigation with Wi-Fi [Online] Available: https://www.infsoft.com/technology/positioning-technologies/wi-fi/ [Accessed November 2021]

[14] [Online] Available: https://www.link-labs.com/blog/ultra-wideband-positioning-location-tracking [Accessed November 2021]

[15] Indoor Positioning, Tracking and Indoor Navigation with Wi-Fi [Online] Available: https://locatify.com/blog/indoor-positioning-systems-ble-beacons/ [Accessed November 2021]

[16] IEEE 2600-2008 - IEEE Standard for Information Technology: Hardcopy Device and System Security [Online] Available: https://standards.ieee.org/standard/2600-2008.html [Accessed November 2021]

[17] IEEE 208-1995 - IEEE Standard on Video Techniques: Measurement of Resolution of Camera Systems, 1993 Techniques [Online] Available: https://standards.ieee.org/standard/208-1995.html [Accessed November 2021]

[18] P2884 - Standard for Performance Evaluation of Biometric Information: Facial Recognition [Online] Available: https://standards.ieee.org/project/2884.html [Accessed November 2021]

[19] P2716 - Guide for the Characterization of the Effectiveness of Printed Circuit Board Level Shielding [Online] Available: https://standards.ieee.org/project/2716.html [Accessed November 2021]

[20] IEC 61188-6-1:2021 Generic requirements for land pattern on circuit boards [Online] Available: https://webstore.iec.ch/publication/62445 [Accessed November 2021]

[21] IEC TR 61191-7:2020 Technical cleanliness of components and printed board assemblies [Online] Available: https://webstore.iec.ch/publication/64835 [Accessed November 2021]

[22] IPC-2615: Printed Board Dimensions and Tolerances [Online] Available: https://shop.ipc.org/general-electronics/standards/2615-0-0-english [Accessed November 2021]

[23] IEC 60068-2-64 Vibration, broadband random and guidance [Online] Available: https://webstore.iec.ch/publication/65892 [Accessed November 2021]

[24] EIA-364: Electrical Connector Performance Test Standards[Online] Available: https://experiorlabs.com/eia-364/ [Accessed November 2021]

[25] IEEE 802.15.1-2002 - IEEE Standard for Telecommunications and Information Exchange Between Systems [Online] Available: https://standards.ieee.org/standard/802_15_1-2002.html [Accessed November 2021]

[26] ISO/IEC 30137-1:2019 Information technology — Use of biometrics in video surveillance systems — Part 1: System design and specification [Online] Available: https://www.iso.org/standard/64935.html [Accessed November 2021]

[27] ISO/IEC 19794-5:2005 Information technology — Biometric data interchange formats — Part 5: Face image data [Online] Available: https://www.iso.org/standard/38749.html [Accessed November 2021]

[28] ISO/IEC 18305:2016 [Online] Available: https://www.iso.org/standard/62090.html [Accessed November 2021]

[29] IEC 62676-5:2018(Main) Video surveillance systems for use in security applications - Part 5 [Online] Available: https://standards.iteh.ai/catalog/standards/iec/ccfabce5-662a-40f9-86d9-3e75fe584795/iec-62676-5-2018 [Accessed November 2021]

[30] Indoor Positioning With Bluetooth Low Energy (BLE) [Online] Available: https://www.instructables.com/Indoor-Positioning-with-Bluetooth-Low-Energy/ [Accessed November 2021]

[31] Eye and face detection using SVM [Online] Available: https://ieeexplore.ieee.org/abstract/document/1417525 [Accessed November 2021]

[32] How to Design a Bluetooth Low Energy Circuit with Sensor Technology [Online] Available: https://www.allaboutcircuits.com/technical-articles/how-to-design-a-bluetooth-low-energy-circuit-with-sensor-technology/ [Accessed November 2021]

[33] “Labeled Faces in the Wild Home.” LFW : Results, vis-www.cs.umass.edu/lfw/.

[34] Step Down Buck Regulator / Converter [Online] Available: https://www.electronics-notes.com/articles/analogue_circuits/power-supply-electronics/switching-step-down-buck-regulator-dc-dc-converter.php [Accessed December 2021]

[35] OpenCV Documentation [Online] Available: https://docs.opencv.org/3.4/ [Accessed November 2021]

[36] OpenCV Object Detection Module [Online] Available: https://docs.opencv.org/3.4/d5/d54/group__objdetect.html [Accessed November 2021]

[37] OpenCV Deep Neural Network Module [Online] Available: https://docs.opencv.org/3.4/d6/d0f/group__dnn.html [Accessed November 2021]

[38] What are Near Field and Far Field Regions of an Antenna? [Online] Available: https://www.everythingrf.com/community/what-are-near-field-and-far-field-regions-of-an-antenna [Accessed December 2021]

[39] Using Alternative Power Sources [Online] Available: https://www.newbiehack.com/MicrocontrollersAlternativePowerSources.aspx [Accessed December 2021]

[40] Beginners guide to the MQTT protocol [Online] Available: http://www.steves-internet-guide.com/mqtt/ [Accessed February 2021]

[41] An Introduction to Socket.IO [Online] Available: https://dev.to/mohamedashrafothman/an-introduction-to-socket-io-3ncb [Accessed February 2021]

[42] A novel trilateration algorithm for RSSI-based indoor localization [Online] Available:https://ieeexplore.ieee.org/document/9036937. [Accessed February 2021]

Appendix C - Datasheets

LM2576xx Series SIMPLE SWITCHER® 3-A Step-Down Voltage Regulator

1 Features

- Newer products available:
  - LMR33630 36-V, 3-A, 400-kHz synchronous converter
  - LM76003 60-V, 3.5-A, 2.2-MHz synchronous converter
- 3.3-V, 5-V, 12-V, 15-V, and adjustable output versions
- Adjustable version output voltage range,1.23 V to 37 V (57 V for HV version) ±4% maximum over line and load conditions
- Specified 3-A output current
- Wide input voltage range: 40 V Up to 60 V for HV version
- Requires only four external components
- 52-kHz fixed-frequency internal oscillator
- TTL-shutdown capability, low-power standby mode
- High efficiency
- Uses readily available standard inductors
- Create a custom design using the LMR33630 or LM76003 with the WEBENCH® Power Designer

2 Applications

- Motor drives
- Merchant network and server PSU
- Appliances
- Test and measurement equipment

3 Description

The LM2576 series of regulators are monolithic integrated circuits that provide all the active functions for a step-down (buck) switching regulator, capable of driving 3-A load with excellent line and load regulation. These devices are available in fixed output voltages of 3.3 V, 5 V, 12 V, 15 V, and an adjustable output version.

Requiring a minimum number of external components, these regulators are simple to use and include fault protection and a fixed-frequency oscillator.

The LM2576 series offers a high-efficiency replacement for popular three-terminal linear regulators. It substantially reduces the size of the heat sink, and in some cases no heat sink is required.

A standard series of inductors optimized for use with the LM2576 are available from several different manufacturers. This feature greatly simplifies the design of switch-mode power supplies.

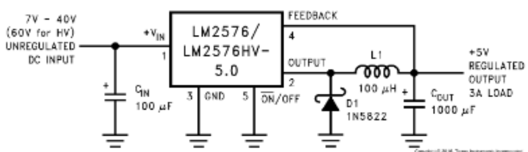
Other features include a ±4% tolerance on output voltage within specified input voltages and output load conditions, and ±10% on the oscillator frequency. External shutdown is included, featuring 50-µA (typical) standby current. The output switch includes cycle-by-cycle current limiting, as well as thermal shutdown for full protection under fault conditions.

The new product, LMR33630, offers reduced BOM cost, higher efficiency, and an 85% reduction in solution size among many other features. The LM76003 requires very few external components and has a pinout designed for simple, optimum PCB layout for EMI and thermal performance. See the device comparison table to compare specs.

Device Information

PART NUMBER(1)	PACKAGE	BODY SIZE (NOM)
LM2576 LM2576HV	TO-220 (5)	10.16 mm × 8.51 mm
	DDPAK/TO-263 (5)	10.16 mm × 8.42 mm

(1) For all available packages, see the orderable addendum at the end of the data sheet.



Fixed Output Voltage Version Typical Application Diagram

Lm2576 Step-Down Voltage Regulator Datasheet