SU1. PROBLEM STATEMENT

The increasing convenience of smart home technologies has transformed the way society interacts with its living spaces. However, current control interfaces often fall short of providing a truly intuitive and inclusive experience. Gestura, a gesture-based smart home system, seeks to address these limitations by introducing a novel approach to interaction.

1.1. Need Statement

According to Kela et al., 76% of people prefer to use gestures over speech when controlling their devices [1]. In today's society, smart home devices are prevalent in homes and businesses. As reported by Statista, 43.8% of households in the US have smart home devices, which is expected to rise to 75.1% by 2028 [2]. Companies like Amazon and Google offer control over smart home devices using voice commands, but no company has invented a gesture-based smart home system. Gesture-based smart home systems would be beneficial to many people, including users with speech disabilities, accents, and those who prefer a quiet household due to noise sensitivities.

1.2. Objective Statement

Gestura is the only smart home product that is controlled by gestures. All other smart home systems are controlled by speech. Gestura links most smart home devices through a central control module and tracks hand gestures via cameras placed throughout a residential or commercial site. Gestura uses a trainable gesture library, offering a seamless experience for the user.

1.3. Background and Related Work

As smart home systems have developed over the years, voice recognition has become the most widespread smart home control type. In 2019, a study conducted by Amazon reported that 70% of the market share was dominated by their Alexa/Echo devices [3]. Gestura is the only gesture-based smart home system available on the market today. However, experiments have been conducted on the efficacy of gesture-based smart home control. For example, an experiment was done by Alemuda and Lin in which they used an accelerometer and gyroscope attached to a bracelet that would provide details about the position of the hand. This information is used to recognize the ten preloaded hand gestures, which are used to control smart home devices like the Hue Smart Light or Bluetooth speakers [4].

2. DESIGN REQUIREMENT SPECIFICATIONS

In the development of Gestura, it is necessary to establish a clear set of design requirements, specifications, and constraints that guide the creation of a user-centric gesture-based smart home system. With safety and reliability as a priority, the design team set strict specifications for the smart home system.

2.1 Requirements

The design team set strict marketing and engineering requirements to ensure ease of use and reliability of the system as a whole. The system accurately tracks gestures and operates efficiently. It is compatible with most smart home devices and suitable for the average home environment.

2.1.1 Marketing Requirements

The marketing requirements are as follows:

1. Gestura allows the user to control their smart home devices via hand gestures.
2. Gestura is easy to install.
3. Gestura reads gestures accurately within 10 feet.
4. Gestura is compatible with most smart home devices.
5. Gestura is suitable in any home environment.
6. Gestura provides user feedback, indicating whether a gesture has been recognized or not.
7. Gestura minimizes the chance of unintentional inputs to the smart home device.

Fig. 2-1 shows an objective tree that aligns with the listed requirements.

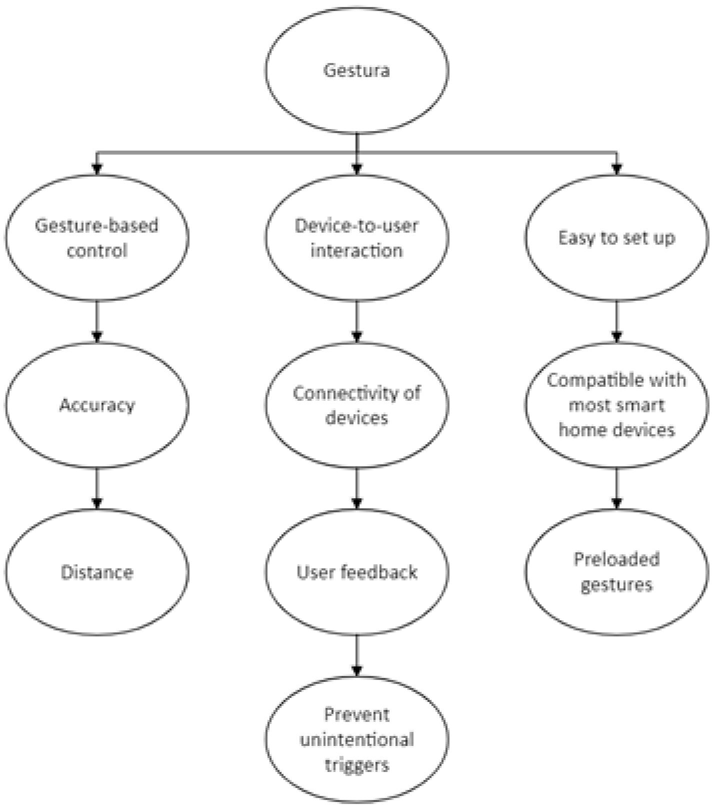

Figure 2-1: Objective Tree for Gestura: Gesture-Based Smart Home

Gestura is designed to maximize user convenience, providing effortless control of the smart home using hand gestures. These marketing requirements not only highlight its user-friendly design but also fowirerm the foundation of the engineering requirements.

2.1.2 Engineering Requirements

Gestura’s engineering requirements are set to enhance the marketing requirements, providing performance and technical specifications. **Table 2-1** is an in-depth look at the team's requirements and a description of why they are needed.

Table 2-1: Engineering Requirements

<table>
<tr><th>Marketing Requirements</th><th>Engineering Requirements</th><th>Description</th></tr>
<tr><td>1, 3, 6, 7</td><td>The camera is high resolution.</td><td>To recognize a hand gesture at 10 ft, the video is high quality.</td></tr>
<tr><td>2, 5</td><td>The device is USB-C powered.</td><td>The device is not complicated to plug in and requires an adequate power supply.</td></tr>
<tr><td>2, 4</td><td>The device takes less than 2 minutes to set up.</td><td>Competitors demonstrate that 2 minutes to power and connect to Wi-Fi is a reasonable time [5].</td></tr>
<tr><td>1, 2, 4, 6</td><td>The device is Bluetooth and Wi-Fi compatible to control the main brands of smart homes.</td><td>Most smart home devices connect through Bluetooth and Wi-Fi.</td></tr>
<tr><td>2, 5</td><td>The dimensions of the device are less than 5in x 6in x 9in.</td><td>The device is compact, thus ensuring that it is easy to set up and place.</td></tr>
<tr><td colspan="3">Marketing Requirements
1. Gestura allows the user to control their smart home devices via hand gestures.
2. Gestura is easy to install.
3. Gestura reads gestures accurately within 10 feet.
4. Gestura is compatible with most smart home devices.
5. Gestura is suitable in any home environment.
6. Gestura provides user feedback, indicating whether a gesture has been recognized or not.
7. Gestura minimizes the chance of unintentional inputs to the smart home device.</td></tr>
</table>

The design team's engineering requirements prioritize ease of use and reliability. This device runs smoothly and easily connects to other devices (lights, thermostats, televisions, and so forth). The device is USB-C powered, which guarantees a reliable power supply and ensures a setup time of less than 2 minutes. The small footprint of the device allows for easy positioning and minimal footprint. The device’s Wi-Fi and Bluetooth capabilities allow for seamless communication with the cameras and smart home devices placed throughout a household. The high-resolution cameras with their 10-foot range make certain that 72% of the average living room is covered and able to recognize gestures [6].

2.2 Constraints

The constraints outline the requirements for Gestura. The constraints are the limitations for the design of the product. These constraints include economics, safety, hardware, and accuracy. Gestura meets all the constraints shown in **Table 2-2**.

Table 2-2: Constraints

Type	Name	Description
Economic	Cost	The budget for the project is $1000 provided by the Department of Electrical and Computer Engineering at Mississippi State University.
Economic	Time	The prototype is functional within two semesters.
Safety	Parental Controls	Parental controls ensure that children cannot control the device without a parent's consent.
Hardware	Power	A USB-C cable is used to charge the device from a standard 120V wall outlet. Many smart home devices are charged via USB-C.
Accuracy	Distance	The device accurately recognizes the gestures within the dimensions of an average-sized U.S. home.

Gestura was designed with a product budget of $1000, and a deadline of two semesters to complete the project. The team decided parental controls are essential for the safe use of the device. For powering the device, a USB-C was chosen to increase efficiency and ease of use. The device guarantees that gestures are recognized and executed effectively across a vast portion of the room where the camera is located. Ideally, gestures are recognized within 10 feet of the device.

2.3 Standards

The design team ensures that the standards in **Table 2-3** are followed for this product. The standards protect the consumer and help the product run more reliably.

Table 2-3: Engineering Standards

Specific standard	Standard document	Specification / Application
USB 3.0	Universal Serial Bus Revision 3.0 specification	This standard defines the capabilities and characteristics of the USB advanced technology. USB 3.0 is used to power the control module.
802.11ac Wireless	Institute of Electrical and Electronics Engineers (IEEE) standard for Wi-Fi	This standard builds upon the foundation of the original Wi-Fi standards. It provides faster and more reliable communication. Wi-Fi is used for the control module to communicate with smart home devices.
2802	Underwriters Laboratories (UL)	This standard is for testing the performance of a camera's image quality by UL performing various tests and giving a performance score to different cameras.
29119-3	International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC)/IEEE	The software test documentation standard is for software testing by the design team keeping various documents to track progress and results while performing tests.

Table 2-3 describes the various standards that are utilized for documenting, testing, and the components used in manufacturing the product and the manufacturing process. The USB 3.0 standard defines the capabilities of advanced USB technology and is implemented for powering the control module. The IEEE standard 802.11ac Wireless is used for wireless communication with smart home devices. UL's standard 2802 ensures the image quality of the camera by using reputable benchmarks from UL. ISO 29119-3 is the third part of the ISO 29119 standard used in software testing, which gives guidance on how the documentation should be written.

3. DESIGN APPROACH

This document describes the design approach for the Gestura product, including design options considered, a system overview, and subsystem descriptions. Gestura is a gesture-based system to control smart home devices. The system is divided into six subsystems: Gesture Recognition, User Interface, Power Supply, Sensor Module, Data Communications, and 3D Design. The product has the following constraints: Economic, Safety, Hardware, and Accuracy. The main requirements for the design include communication and accuracy. The product budget is $1000. The product ensures parents that the device cannot be controlled without parent's consent. The hardware of the device is customer-friendly and easy to assemble. The device is accurate within the dimensions of an average-sized U.S. room.

3.1 Design Options

For Gestura, the main challenge was how to collect data for gesture recognition. Two main design choices were considered. The first design option used a wristband that the user would wear to track the movement of the hand. The other approach involved using cameras and AI to detect gestures.

3.1.1 Design Option 1

The first design option considered was using a wristband with a gyroscope and an accelerometer that collected data to send to a microcontroller over Wi-Fi, which would interpret the data and signal an output to the smart home device. The advantages of this approach were that it was less expensive to make and provided more accurate data, but the disadvantage was that wearing a wristband could be uncomfortable or inconvenient for users. This approach was done in the Institute of Electrical and Electronics Engineers (IEEE) study by Alemuda and Lin [4].

3.1.2 Design Option 2

The option the team chose was to use computer vision instead of a wristband to recognize gestures. This can be done through a machine learning algorithm to recognize certain gestures over others and send smart home commands when a gesture is recognized. Two advantages are that this approach is user-friendly and that there is no extra technology that the user must wear. On the other hand, the disadvantages are that it is very costly, and extracting valuable data is much more difficult. The design team decided to tackle the second approach because the comfort and accessibility to the user in this approach outweigh the benefit of the low cost in the first approach.

3.2 System Overview

Fig. 3-1 shows how Gestura works at a glance.

Fig. 3-1: Gestura System at a Glance (Level 0)

Gestura works by taking in a gesture through a camera. Gestura recognizes the gesture and outputs a command to a smart home device. The system works through wireless communications via Wi-Fi. All communications go through the control module. **Fig. 3-2** is an in-depth view of the system.

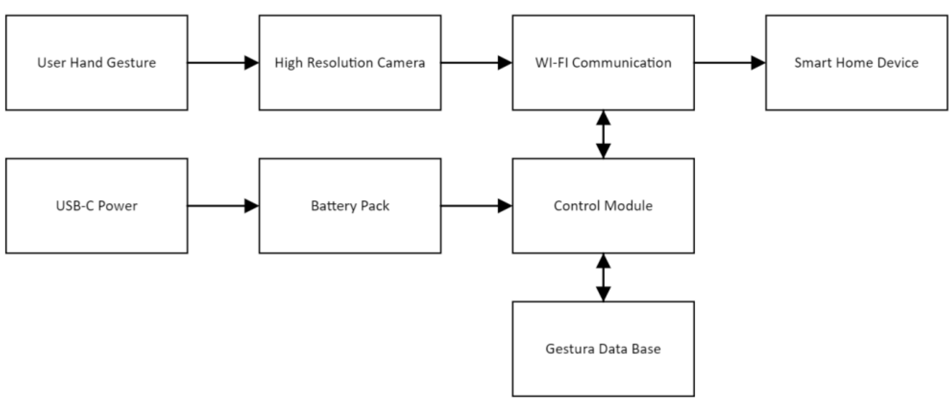

Fig. 3-2: Gestura System Diagram (Level 1)

The user's hand is captured through a camera and sent to the control module. The control module analyzes the video to compare it to its gesture library. The module outputs a command that is sent over Wi-Fi to the smart home device. This interaction is explained more in the following subsystem descriptions.

3.2.1. Microprocessor

For this project, the team considered two different options for the microprocessor. The first option was the Raspberry Pi 4 Model B, while the second option was an Allen Bradley programmable logic controller (PLC). **Table 3-1** provides the comparisons.

Table 3-1: Advantages vs. Disadvantages

Microprocessor	Price	Wireless connectivity	Supported Programming Languages
Allen Bradley PLC [7]	$255.00	None	Ladder Logic
Raspberry Pi 4 Model B [8]	$89.99	Built-in Bluetooth and Wi-Fi	Python, C, C++, Java, and many more

The Raspberry Pi offers multiple advantages compared to the PLC. First, the Raspberry Pi is more cost-effective. The Raspberry Pi is $165 cheaper than the PLC. Secondly, the Raspberry Pi can use multiple different programming languages, while the PLC can use only one. Additionally, for the PLC to be the team's main control module, it required extra products to connect to the team's different subsystems. The Raspberry Pi, however, already has built-in Wi-Fi and Bluetooth connectivity. Given that the team's project focused on residential use, the team found the PLC exceeded the project's needs, and the Raspberry Pi met the project's specific needs more effectively. **Fig. 3-3** is a diagram of the team's chosen microprocessor.

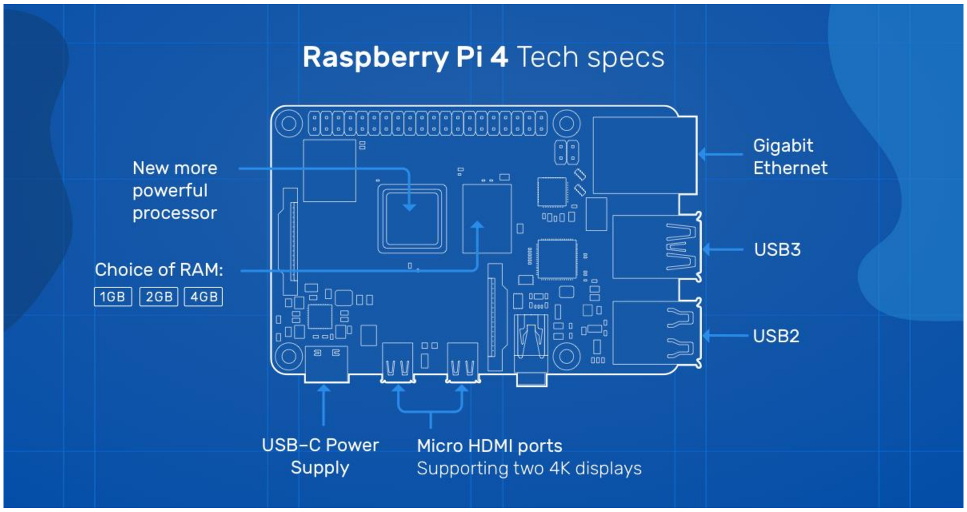

Fig. 3-3: Raspberry Pi 4 Model B [8]

The team chose to have 8 GB of RAM for the Raspberry Pi because of the importance that RAM has in AI and computer vision.

3.3 . Subsystems

Gestura has six subsystems. Subsystem 1 is the gesture recognition subsystem. It detects and analyzes the gestures the user uses. Subsystem 2 is the user interface subsystem. The interface is on a tablet to allow the user to interact, connect smart home devices, and display user feedback. Subsystem 3 is the power supply subsystem that powers the LCD screen and Raspberry Pi for the tablet. Subsystem 4, the sensor module subsystem, detects and sends the information of the gestures to the Raspberry Pi. Subsystem 5 is the data communications subsystem, passing data among the control module, the smart home devices, and the cameras. The final subsystem is the 3D design subsystem. This 3D enclosure holds the Raspberry Pi, fan, LCD screen, power supply, and internal wiring for the tablet.

3.3.1. Gesture Recognition Subsystem

The gesture recognition subsystem is responsible for detecting gestures by using computer vision and analyzing the gestures using a machine learning algorithm to assign them to a certain function. Popular programming languages for computer vision and machine learning are C++ and Python [9]. **Table 3-2** shows the options.

Table 3-2: Programming Language Options

Programming Language	Pros	Cons
Python	Easy to use and expansive library for computer vision	Slower
C++	Faster	Harder to use and limited libraries on computer vision
MATLAB	Suited for computer vision	Lack of experience and limited libraries

The design team chose Python because of its extensive library [10] and the experience that the design team has with the programming language. **Fig. 3-4** shows a general flowchart of the gesture recognition subsystem.

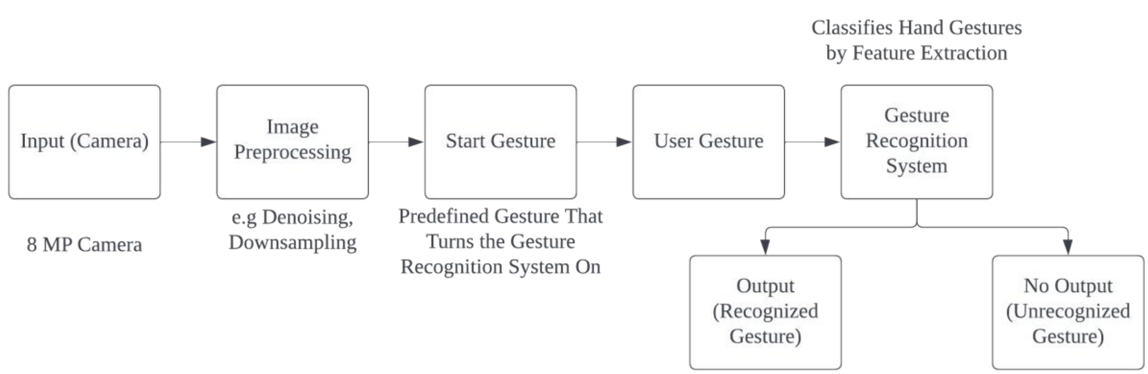

Fig. 3-4: Gesture Recognition Flowchart

An 8 MP camera was chosen because an increase in the quality of the video input results in a more accurate output in vision-based AI. As reflected by Cheng, Gestura needs high-quality input to be effective [11]. Image preprocessing, such as denoising and downsampling, is used to reduce the file size, which increases the efficiency and speed of gesture recognition. The “start gesture” is a gesture stored on the device that makes Gestura look for user gestures once the gesture is performed. This implementation was done to prevent unintentional activation of the device. User gestures are gestures that, once the confidence threshold for the gestures has been reached, activate the smart home device that they are programmed to affect. Training data is stored on the device with predefined gestures. The team is exploring adding user-programmable gestures next semester.

3.3.2. User Interface Subsystem

The user interface is on a tablet powered by the Raspberry Pi. The touch screen allows the user to connect to a home Wi-Fi network or new smart home devices. It also outputs if a gesture is detected and the corresponding command. **Fig. 3-5** offers a visual depiction of the user interface.

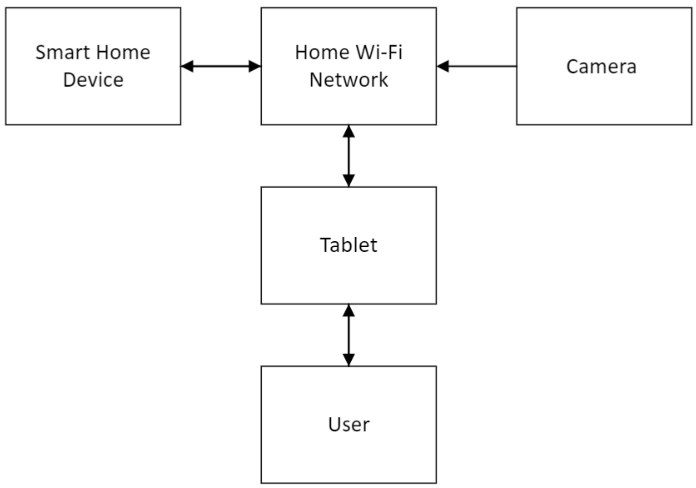

Fig. 3-5: User Interface Diagram

Amazon offers several different touch screens with resolution quality and sizes based on cost. The team decided that based on resolution and cost, the Hosyond 7-inch liquid crystal display (LCD) touch screen offers ample size and quality at a lower price compared to other options. The options are listed below in **Table 3-2**.

Table 3-3: Touch Screen Options

Product Description	Price	Size	Power Dissipation	Refresh Rate	Built-in Speakers
Hosyond 3.5 Inch 480x320 Touch Screen TFT LCD SPI [12]	$17.99	4.61 x 2.8 x 1.34 inches	0.65 W	10 Hz	No
Hosyond 7 Inch IPS LCD Touch Screen Display Panel 1024×600 [13]	$45.99	6.5 x 5 x 0.6 inches	3.1 W	60 Hz	No
GeeekPi 7 Inch Capacitive Touch Screen 1024x600 IPS LCD Display [14]	$59.99	6.65 x 3.93 x 0.32 inches	3.1 W	60 Hz	Yes

Although the Hosyond 3.5-inch touch screen is inexpensive at $17.99 and consumes only 0.65 W of power, its size and refresh rate make it difficult to use. Two viable options were left. The Hosyond 7-inch touch screen offers many of the same features as the GeeekPi 7-inch touch screen at a lower price apart from built-in speakers, which are not needed for the project, so the Hosyond 7-inch touch screen was chosen for its lower price.

3.3.3. Power Supply Subsystem

The Raspberry Pi and the LCD touch screen are powered by a 5-V rechargeable power supply. The power supply uses four 18650 batteries in series to achieve the longevity of the device. The device has 7000 mAh, and the power supply can be recharged by a universal serial bus (USB) Type-C charger. The cameras have a separate power supply. **Fig. 3-6** shows a general diagram of the power supply subsystem.

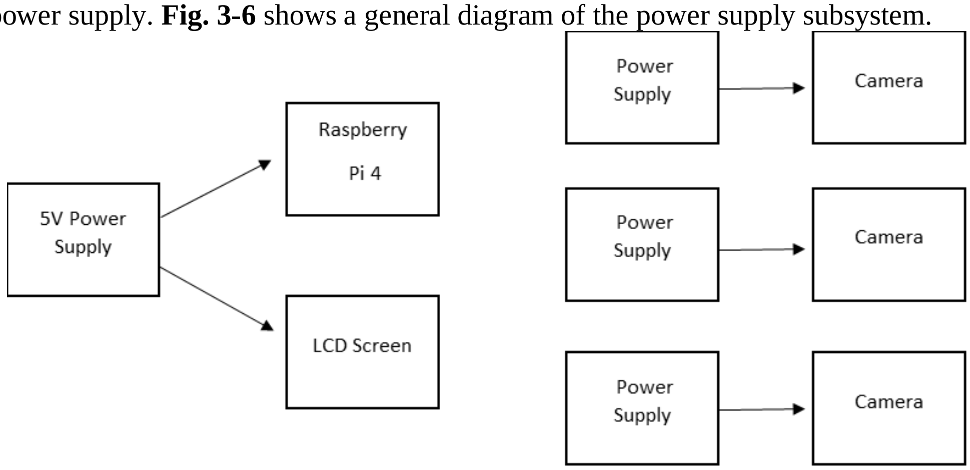

Fig 3-6: Power Supply Diagram

The voltage required by both the screen and the controller is 5 V. The team chose a rechargeable power supply to eliminate the hassle of replacing batteries. With the camera constantly monitoring for gestures, a long-lasting battery life is needed for the device. The team chose four 18650 batteries in series to achieve 7000 mAh. With 7000 mAh, the device can run for two to three days with moderate use before needing a charge.

3.3.4. Sensor Module Subsystem

The sensor module subsystem is responsible for taking in the input from the cameras of the system. This is done by using Wi-Fi to retrieve footage from the cameras, which is sent to the gesture recognition subsystem. **Fig. 3-7** below shows the flow chart from the input to the output of the subsystem.

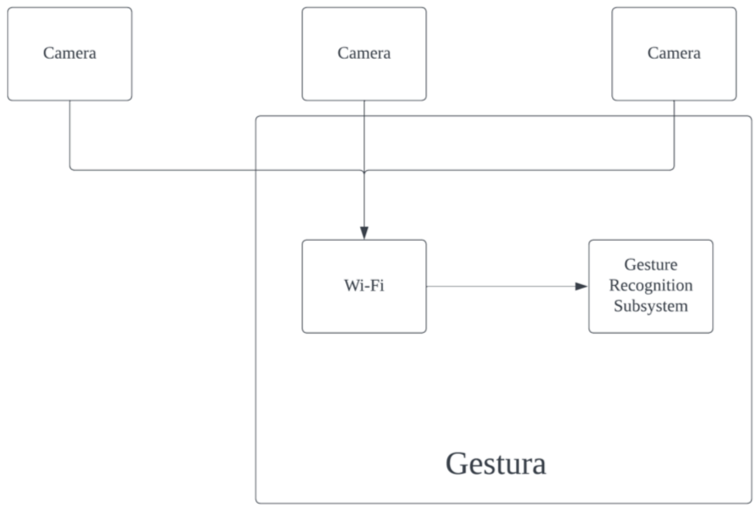

Fig. 3-7: Sensor Module Flowchart

The code is in Python using the CV2 library. This library takes input of each frame of the recording of the camera. This method can be used for multiple cameras. Each is sent to the gesture recognition subsystem to look for a start gesture. Once the start gesture is recognized by a camera, Gestura takes input from only that camera. **Fig. 3-8** demonstrates this interaction.

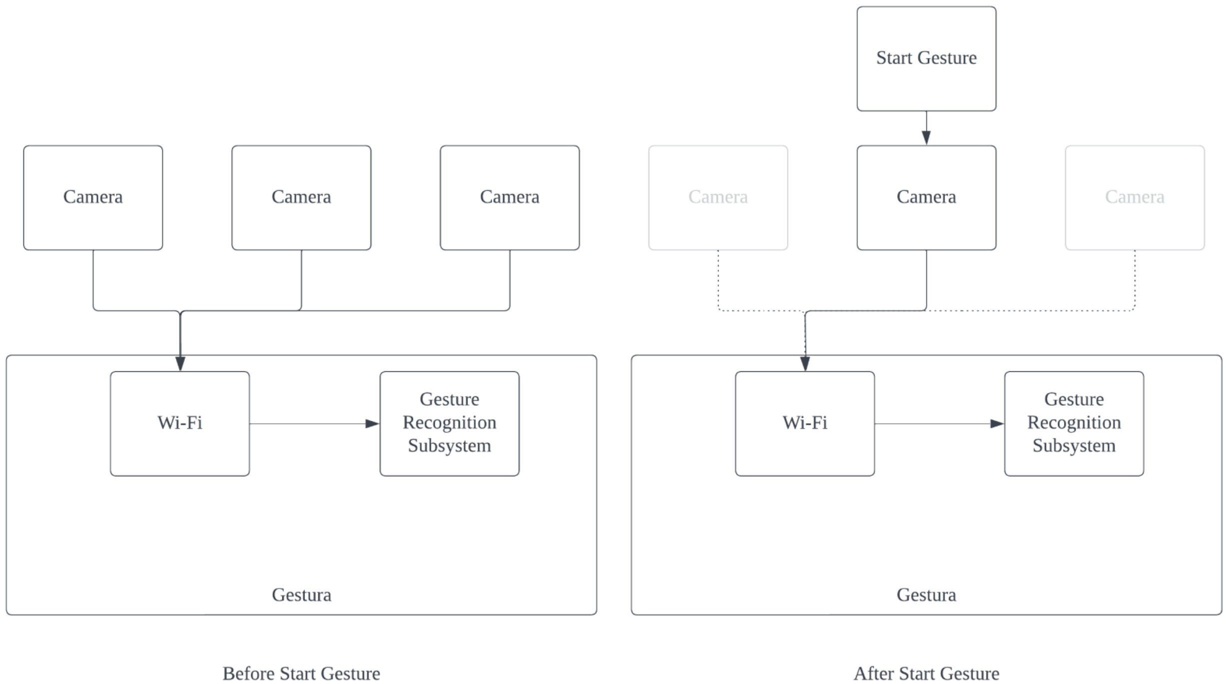

Fig. 3-8: Start Gesture Interaction

This implementation speeds up the response time of Gestura by efficiently using processing power where it is important.

3.3.5. Data Communications Subsystem

Wi-Fi is used to communicate with smart home devices and the server. This is due to Wi-Fi being the most common pairing method for smart home devices. **Fig. 3-9** provides a general overview of the data communications subsystem.

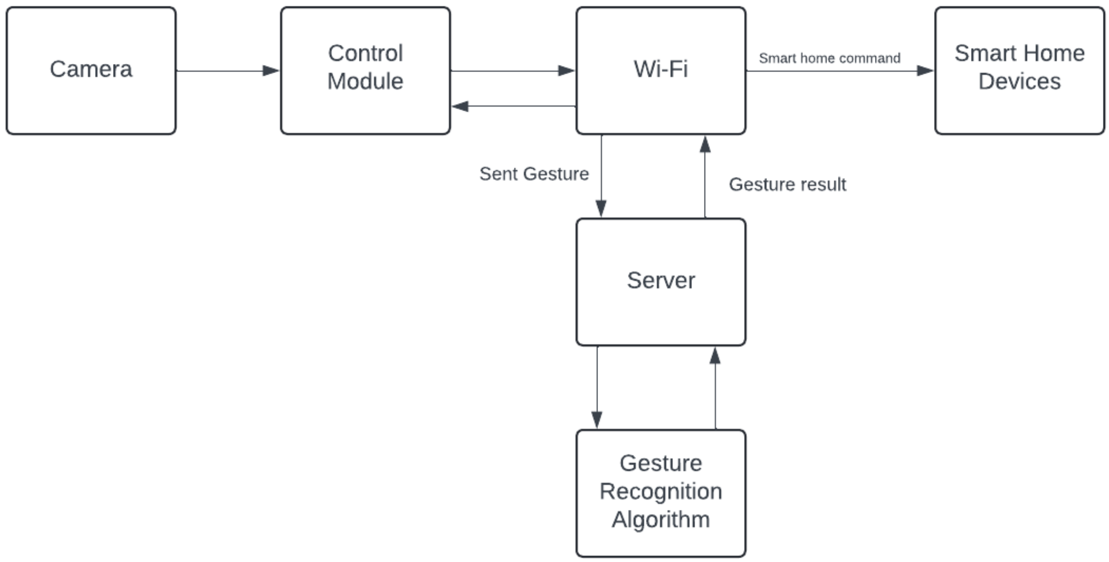

Fig. 3-9: Data Communications Diagram

The server and smart home devices use the home network to communicate with the control module. Once the returned gesture result has been sent from the gesture recognition algorithm, it is sent over Wi-Fi to the control module, which translates the signals into usable commands.

3.3.6. 3D Design Subsystem

The 3D design subsystem holds the Raspberry Pi, fan, LCD screen, power supply, and internal wiring in an enclosure for the tablet. The ports of the Raspberry Pi are also accounted for in the design by leaving open holes to access the ports easily and to plug in wiring. **Fig. 3-10** displays all components that are part of the 3D design subsystem.

Fig. 3-10: 3D Design Diagram

The enclosure is designed to hold the components that are in **Fig. 3-10.** Its design is in the form of a tablet to make it easy to handle and use for the user.

3.4. Level 2 Prototype Design

For Gestura, the team is planning to encase the control module, cables, battery pack, battery charger module, and so forth in a 3D-printed case. The case has a touch screen monitor on it to act like a tablet. The team intends to use acrylonitrile butadiene styrene (ABS) filament to make the case. ABS plastic is a sturdy enough material for the case to endure moderate wear and tear.

3.4.1. Level 2 Diagram

Building upon the Level 0 and Level 1 diagrams previously shown, a diagram that shows how all these subsystems work together is shown in **Fig. 3-11**.

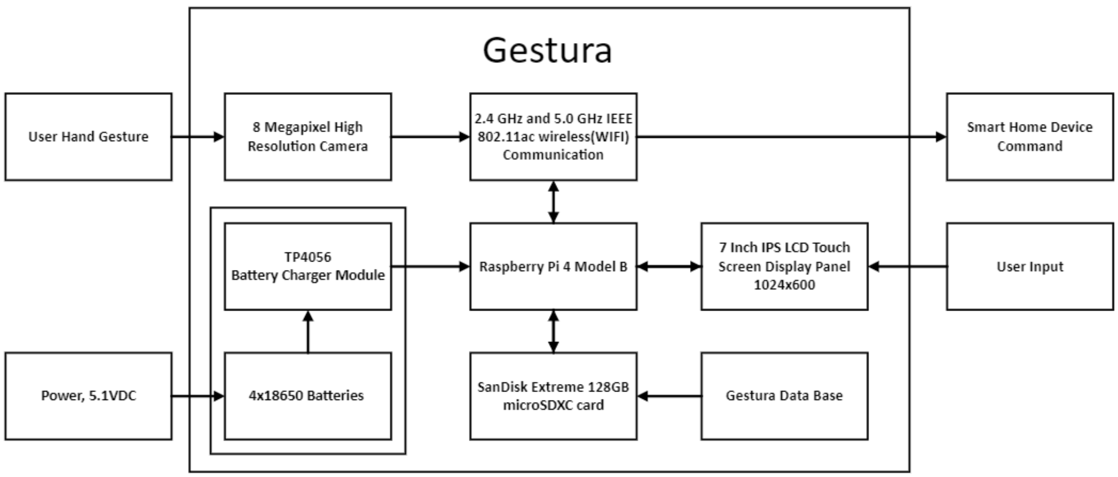

Fig. 3-11: Diagram for Gestura (Level 2)

Each subsystem provides a vital role in Gestura's functionality. The choices made herein provide a foundation of how Gestura works. The next step is to validate these concepts with rigorous testing, addressing any problems as they arise.

4. EVALUATION

This section covers the testing process that was used for Gestura. Each section covers the methods that were used for a subsystem to ensure it aligns with the project's requirements and constraints.

4.1. Testing Certification – Subsystems

Gestura, a gesture based smart home system, is comprised of 5 different subsystems. They are as follows: user interface, microprocessor, data communications, gesture recognition, and power supply. These tests were created to test the effectiveness of each subsystem. It is important to make sure each subsystem works individually before testing if all the subsystems work as one system. Each subsystem has a requirement specification that must be passed before moving on to further testing. The subsystems were in categories such as responsiveness, accuracy, or intuitiveness.

4.1.1. User Interface

The user interface subsystem is responsible for letting the user interact with the system. It allows them to connect to new smart home devices and Wi-Fi networks, see recent gesture commands and set up which gestures give which commands. For this test, the design team had 10 participants use the user-interface for 5 minutes and then give a score from 1 to 5 on the intuitiveness of the layout. **Table 4-1** shows the results.

Table 4-1: Results of User Interface Survey

User Interface Survey results					
Score	1	2	3	4	5
Number of participants	0	1	7	2	0

Based on the results of the test, our user interface could be improved. Most of the votes say that the user interface is usable but not preferable. This will lead to the design team reiterating tests on the user interface in the future.

4.1.2. Microprocessor

The microprocessor is one of the most important components of the device. If the microprocessor fails, the whole product will not work. Therefore, it is important tests are done to ensure that it is not overloaded and that the load does not cause the microprocessor to overheat. In the first test in **Table 4-2**, a test was performed for the CPU usage while under load at idle and during CPU-intensive tasks expected for the device's everyday usage.

Table 4-2: Test Sheet for Microprocessor

CPU usage at idle	CPU usage with regular load
~21%	~73%

Based on the results of this test, the product should not be overloaded even under load that will be expected during gesture recognition. This will ensure that there is no lag, which will help with the device's usability and that the chance of successful gesture recognition is not reduced because of throttling.

4.1.3. Data Communications

As the server response time is part of the time that the user must wait, it is paramount to investigate how to potentially reduce time for the server to respond and, as a result, make the responsiveness of the product faster as our requirements outline. The first test in **Table 4-3** is to measure the average response time for the server to take the images from the camera and send back a gesture result to the user.

Table 4-3: Results of the Data Communications Test

	Server Response Time	Server Sending Time
Times (s)	~0.251	~2.315

The result of this test shows that the server response time and sending time will not have too big of an impact on the response time of the device which will make the user not have to wait an unreasonable amount of time for processing.

4.1.4. Gesture Recognition

The gesture recognition software is how Gestura recognizes hand gestures. **Figure 4-1** shows the results of testing the speed and accuracy of Gestura at various distances.

Test Writer: Eric Duncan			
Test Case Name:	Gesture recognition test	**Test ID #:**	
Description:	Determine the accuracy and time of gesture recognition at various distances.	**Type:**	☐ white box ☑ black box
Tester Information			
Name of Tester:	Eric Duncan	**Date:**	02/26/2024
Hardware Ver:	N/A	**Time:**	5:00 PM
Setup:	The camera is set up facing the user		

Test	**Distance (feet)**	**Time (s)**	**Accuracy**	**Pass**	**Fail**	**N/A**	**Comments**
1	2	1.4	90%	☑			9/10 gestures successfully recognized
2	5	2.7	80%	☑			8/10 gestures successfully recognized
3	10	3.5	60%	☑			6/10 gestures successfully recognized
	Overall test result:			☑			23/30 gestures successfully recognized

Figure 4-1: Test for Gesture Recognition

The test has the camera facing the user to measure the accuracy of the gesture recognition algorithm at different distances. This test was conducted directly on the server that handles gesture recognition to mitigate inaccuracies stemming from internet connection problems. A timer was coded into the program to get accurate times of how long each run took. This test had three sections: 2, 5, and 10 feet away. Each section had 30 trials with 3 people trying the same gesture for 10 trials. The calculation for accuracy was calculated by dividing the number of successful inputs by the number of misinputs or unrecognized results. The results of this test demonstrate that Gestura meets its marketing requirements for accuracy within a 10-foot range.

4.1.5. Power Supply

The Power supply for Gestura should always be 5V and should last up to 24 hours on one charge. The test will be conducted by starting with 5V and using a timer to see how long the voltage remains at 5V. When the voltage drops below 5V the timer will be stopped. The results from this test will determine the longevity of the power supply and determine any changes that need to be made to satisfy the criteria. **Figure 4-2** shows the results of the test.

Test Writer: Shenna Booker							
	Test Case Name:		Power Supply Testing	**Test ID #:**			
	Description:		Determine how long the battery will last	**Type:**			☐ white box ☑ black box
Tester Information							
	Name of Tester:		Shenna Booker	**Date:**			2/19/2024
	Hardware:		Power Supply & Timer	**Time:**			3:00 PM
	Setup:		The device was left on at an idle state				
Test	**Starting Voltage**	**Final Voltage**	**Time**	**Pass**	**Fail**	**N/A**	**Comments**
1	5V	5V	24Hrs	☑			
	Overall test result:			☑			

Figure 4-2: Power Supply Testing for Gestura

The results of this section show that the device has a long-lasting battery life and can last for up to the marketing requirement of 24 hours on one charge. This ensures that our product will be convenient to use as it will not require a charge for a long time.

4.2. Testing Certification – System Testing

This is the beginning of the comprehensive system testing phase, which is a crucial step for validating our full prototype. System testing is crucial for making sure the entire system works correctly, meets the design specifications, and does what it is supposed to do. Unlike testing individual parts, system testing looks at how everything works together, including its reliability and how it interacts with other systems.

During system testing, the design team will use different methods to check everything. They will do functional testing to make sure each part of the software works correctly according to the requirements. Then, they will do performance testing to see how well the system handles different workloads. They will also do usability testing to check if the system is easy to use. Finally, they will do security testing to keep everything safe and follow the right standards.

Throughout this phase, the design team will carefully document all the tests, results, and any problems they find. This helps them fix any issues before the system is used in the real world.

4.2.1 Gesture Recognition Subsystem

The gesture recognition subsystem includes the software and hardware involved with recognizing a gesture. This includes the integration of the microprocessor, the data communications, the gesture recognition software, and the power supply subsystems. For this test, the device will be operating on the power supply on the Raspberry Pi 4 using an ethernet cable for the internet connection and connecting to a server hosted on Google Cloud. The test was set up in a comparable way to the Gesture Recognition Software test to compare. **Figure 4-3 s**hows the results of the test.

Test Writer: Eric Duncan							
Test Case Name:		Gesture recognition subsystem test		**Test ID #:**			
Description:		Determine the accuracy and time of gesture recognition at various distances using all the parts of the device		**Type:**			☐ white box ☑ black box
Tester Information							
Name of Tester:		Eric Duncan		**Date:**			03/29/2024
Hardware Ver:		N/A		**Time:**			3:00 PM
Setup:		The camera is set up facing the user					
Test	**Distance (feet)**	**Time (s)**	**Accuracy**	**Pass**	**Fail**	**N/A**	**Comments**
1	2	3.1	80%	☑			8/10 gestures successfully recognized
2	5	3.7	70%	☑			7/10 gestures successfully recognized
3	10	4.3	70%	☑			7/10 gestures successfully recognized
Overall test result:				☑			22/30 gestures successfully recognized

Figure 4-3: Test for Gesture Recognition Subsystem

From the results of the gesture recognition subsystem for integration, Gestura has reached the goal of having a 50 percent success rate at 10 feet away as well as the time being under 5 seconds, which matches our marketing requirements and passes the test.

4.2.2 Smart Home Connection System

The smart home connection system will be used to send commands to the smart home systems that are supported by the device. The test will turn on and off a Kasa Smart Plug using the integrated design. The server is run on Google Cloud and the Kasa Smart Plug will be handled on the Raspberry Pi 4. **Figure 4-4** shows the result of this test.

Test Writer: Eric Duncan							
	Test Case Name:	Smart Home Connection System Test		**Test ID #:**			
	Description:	Determine the speed and accuracy of Gestura under normal operations.		**Type:**			☐ white box ☑ black box
Tester Information							
	Name of Tester:	Eric Duncan		**Date:**			2/19/2024
	Hardware:			**Time:**			3:00 PM
	Setup:	A Kasa smart plug is plugged into the wall and is currently connected with the device. The server is up and running. This is repeated for 10 trials.					
Test	**Distance (ft)**	**Accuracy**	**Time until Kasa Smart Plug turns on**	**Pass**	**Fail**	**N/A**	**Comments**
1	2	90%	4.1 s	☑			9/10 Gestures Recognized
2	5	80%	3.9 s	☑			8/10 Gestures Recognized
3	10	60%	4.5 s	☑			6/10 Gestures Recognized
	Overall test result:			☑			

Figure 4-4: Test for Smart Home Connection System

From the data of these tests, this shows that Gestura can process gestures at 10 feet in its current state and can send commands to smart home devices in a timely manner. The integration of each subsystem is complete, and it meets the design team's requirements.

Using the data collected from these tests, the design team ensures that Gestura's subsystems are ready for integration and identifies areas of improvement. Rigorous analysis was used in each test to find the strengths and weaknesses within the subsystems.

Entering the final segments of the document, the design team reflects on the project: its lessons, its goals, and its future. This section serves not as an end but as a beginning to new endeavors.

5. SUMMARY AND FUTURE WORK

Considering the time constraints and lack of knowledge of how to implement this project that we had at the beginning, we are satisfied with the progress we made with this project. The project has taught our team many lessons on team management and how to make a product of our own. While some design team members wish to leave the project for job searching, others want to improve and refine it to make it a product that can be used to make a business. In this case, a set of goals to reach before making a business include making custom gestures possible, increasing the compatibility of smart home devices, and increasing the accuracy of gesture recognition at 10 feet to at least 90 percent.

6. ACKNOWLEDGEMENTS

We would like to acknowledge Dr. Edward for his knowledge that helped steer us in the right direction with this project. We would also like to acknowledge Dr. Beck for encouraging us to go forward with this idea.

7. REFERENCES

[1] J. Kela et al., "Accelerometer-based gesture control for a design environment - personal and ubiquitous computing," SpringerLink, https://link.springer.com/article/10.1007/s00779-005-0033-8#citeas (accessed Sep. 29, 2023).

[2] "Smart home - United States: Statista market forecast," Statista, https://www.statista.com/outlook/dmo/smart-home/united-states (accessed Sep. 29, 2023).

[3] “Intriguing Amazon Alexa Statistics You Need to Know in 2023,” Safeatlast, https://safeatlast.co/blog/amazon-alexa-statistics/#gref (accessed Sep. 29, 2023).

[4] F. Alemuda and F. J. Lin, "Gesture-Based Control in a Smart Home Environment," 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Exeter, UK, 2017, pp. 784-791, doi: 10.1109/iThings-GreenCom-CPSCom-SmartData.2017.120.

[5] “Set up Alexa in a Few Easy Steps,” Amazon, https://www.amazon.com/alexa-setup-guide/b?ie=UTF8&node=17978645011#:~:text=In%20about%20a%20minute%2C%20the,and%20Alexa%20will%20greet%20you (accessed Sep. 29, 2023).

[6] “Average Living Room Size And How To Make Yours Look Larger,” Homedit, https://www.homedit.com/average-living-room-size/#:~:text=A%2012%20x%2018%20ft,need%20a%20larger%20living%20room (accessed Sep. 29, 2023)

[7] “PLC hardware - Allen Bradley 2080-LC30-24QBB series B, surplus in ...,” PLC Hardware, https://www.plchardware.com/Products/RA-2080-LC30-24QBB-B-NSS.aspx (accessed Oct. 24, 2023).

[8] Raspberry Pi, “Raspberry Pi 4 Model B specifications,” https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/ (accessed Oct. 23, 2023).

[9] Y. Noema, “Top 3 programming languages for implementing a computer vision system,” Medium, https://medium.com/imagescv/my-top-3-programming-languages-for-implementing-a-computer-vision-system-2aa0a3ad0a2a (accessed Oct. 20, 2023).

[10] M. R. Munawar, “C++ vs. Python: Choosing the best language for computer vision development,” VisoByte, https://www.visobyte.com/2023/07/cpp-vs-python-choosing-the-language-for-computer-vision-development.html (accessed Oct. 20, 2023).

[11] D. (Wei-Y. Cheng, “Ai with high IQ (image quality).,” Ambarella, https://www.ambarella.com/blog/ai-with-high-iq/ (accessed Oct. 22, 2023).

[12] “Hosyond 3.5 inch 480X320 touch screen TFT LCD SPI display ...,” https://www.amazon.com/Hosyond-480x320-Screen-Display-Raspberry/dp/B0BJDTL9J3 (accessed Oct. 22, 2023).

[13] “Hosyond 7 Inch IPS LCD Touch Screen Display Panel 1024×600 Capacitive Screen ...,” https://www.amazon.com/Hosyond-Display-1024%C3%97600-Capacitive-Raspberry/dp/B09XKC53NH (accessed Oct. 22, 2023).

[14] “GeeekPi 7 inch LCD touch screen for Raspberry Pi, 1024x600 ...,” https://www.amazon.com/GeeekPi-Raspberry-1024x600-Display-Portable/dp/B0CJWXWJ6K (accessed Nov. 3, 2023).