ASHESI UNIVERSITY

DESIGN OF HEALTH MONITORING AND ANALYSIS SYSTEM FOR PETS (CATS) - KELPET.

CAPSTONE PROJECT

BSc. Computer Engineering

Jacinta Esi Amoawah Badu

Class of 2024

ASHESI UNIVERSITY

DESIGN OF HEALTH MONITORING AND ANALYSIS SYSTEM FOR PETS (CATS) -KELPET.

CAPSTONE PROJECT

The capstone project was submitted to the Department of Engineering at Ashesi University in partial fulfillment of the requirement for the Bachelor of Science in Computer Engineering award.

Jacinta Esi Amoawah Badu

Class of 2024

Supervisor: Mr. Kofi Adu-Labi

DECLARATION

At this moment, I declare that this capstone is the result of my original work and that no part has been presented for another degree in this university or elsewhere. Candidate's Signature:

………………………………………………………………………………………

Candidate's Name: ……………………………………………………………………

Date: ………………………………………………………………………………………

I now declare that the preparation and presentation of this capstone were supervised per the guidelines for the supervision of the capstone laid down by Ashesi University College.

Supervisor's Signature:

………………………………………………………………………………………

Supervisor's Name:

………………………………………………………………………………………

Date: ………………………………………………………………………………………

Acknowledgments

I want to express my deepest gratitude to all those who supported me throughout my capstone journey. First, I am profoundly grateful to God for granting me the wisdom, strength, passion, and perseverance to complete this project. I extend my heartfelt thanks to my supervisor, Mr. Kofi Adu-Labi, whose guidance, insights, and encouragement were invaluable. My gratitude also extends to the Faculty of Engineering at Ashesi University for their technical support. Additional appreciation goes to Mr. Nicolas Tali for his technical advice.

Additionally, I would like to acknowledge Prof. Nathan Amanquah for his support and insights on the project implementation. Thanks to Dramani Alhassan and my family for their unwavering support, understanding, patience, and technical advice throughout this journey. Your constant encouragement has been a source of strength and motivation. I would also like to acknowledge my colleagues and peers for their collaboration and constructive feedback. Your input has been vital in refining my work.

Lastly, I sincerely thank my pet [Kelvin] for giving me the motivation and reason to do this project and to everyone who has directly or indirectly contributed. Your support and belief in me have been pivotal in bringing this project to fruition.

Thank you all for your invaluable contribution.

Abstract

Pet health is a concern for most pet owners. Recently, the number of pets has increased, and the most popular ones among us are cats and dogs. They serve as companions. These pets have become part of most people's families, whether in Africa or outside Africa. However, it was documented that about 98% of households have pets as family. However, COVID-19 greatly impacted pets' health, as most pet owners needed help sending their sick pets to veterinary hospitals. This problem has shown that there is a need for a smart health system for pets. However, the issue associated with such devices in Africa, especially Ghana, is that they are rare. Currently, many importation fees and levies must be paid for shipping. Therefore, it has contributed to the discouragement of such device usage and the costly price of the devices. Thus, a low-cost smart pet health monitoring system is needed. The system should be able to analyze and monitor pet health. In this project, a smart pet collar was designed to monitor the pet's heart rate, temperature, activity level, and GPS location, in addition to a mobile app for monitoring these metrics. The software had an online notifier informing pet owners about their health. Health tips to guide them on the things to do as they try to contact a veterinarian and a map to locate all veterinary hospitals in their area. The app also allows pet owners to add up to 5 pets of their choice to monitor their separate health, but the owner must have five different smart pet collars. The owner needs the hardware serial number before they can add more pets. This project stands out from the existing ones because it is less costly with more features than the existing ones. The device with the mobile app comes at $85 yearly, whereas the existing ones cost $600 plus annually, excluding transportation. The product offers pet owners satisfaction and versatility for their health.

Keywords: Kelpet, cat, smart pet collar, mobile application, PCB board, SolidWork.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Symbols	Names
LiPo Battery	Lithium Polymer Battery
P	Input power
I	Operating current
V	Operating voltage
HR	Heart rate
HRV	Heart Rate variability
KelPet	Overall project name
I^2C	Inter-Integrated Circuit
QR Code	Quick Response Code
KNN	K-Nearest Neighbor
PCB	Printed Circuit Board

Chapter 1: Introduction

1.1 Background

A common and age-old problem is that pets are prone to overeat, live an inactive lifestyle, and do other things that may shorten their already short lives [1]. Pet owners are prone to overfeeding and pampering their pets without an incentive to do otherwise, which adds to that shortening. A system is needed to improve habits, activities, and other facets of health, such as heartbeat, respiratory rate, etc. In the United States of America, pets are cherished members of households, with approximately 66% of households having at least one pet [1]. Among these pets, cats hold a significant place, with 76% of cat owners considering their feline companions as family members [1].

Additionally, the percentage of dog and cat ownership in the United Kingdom is comparatively the same as in Ghana. However, these pets are mostly kept outdoors in Ghana [5]. Research on pet ownership trends revealed that while dogs are the most owned pets, cats are a close second in popularity [3]. This growing trend of cat ownership underscores the importance of developing advanced technologies to monitor and analyze cat health effectively. The pet (cat) health monitoring system and analysis project aims to address the specific needs of pet owners, providing them with real-time insights into their pet's health and behavior.

1.2 Problem Statement:

According to Grand Valley Animal Hospital, pets are exceptionally good at hiding their discomforts [4]. While some signs are apparent, such as extreme weight loss, seizures, or bloody stool, others, like heart rate, temperature level, and so on, are so subtle that they can go unnoticed by their human caretakers.

Although most pet owners take their pets to the vet annually to detect potential health risks, sometimes constant monitoring is needed to observe behavioral changes. In addition to visual observation, incorporating technology can accurately monitor their health, allowing for timely actions to alleviate pain. Cats are the pets most likely to hide their illnesses [9]. Lack of awareness can delay treatment, affecting their well-being and increasing veterinary costs.

Based on observations and interactions with some Ghanaians, I realized that they primarily use traditional methods to determine their pet's health, as these smart pet devices are not produced in Africa. The cost of these devices is high due to transportation and various processes [5], which discourages owners from acquiring these devices for their pets.

To address this issue, an efficient, low-cost, non-intrusive, and user-friendly pet health monitoring and analysis system is needed to continuously collect, analyze, and visualize pet health information, providing timely insights to pet owners.

1.3 Project Objectives

Design and implement the pet health monitoring system and analysis for non-stray cats. The summarized objectives are as follows:

1. Design a system that monitors pet health (such as heart rate, activity level, pulse rate, and Heart Rate Variability (HRV)) with or without supervision.
2. Be able to measure and analyze the data for immediate action.
3. Develop a mobile or web app for pet owners to visualize health information, such as the pet's HRV.
4. Alert pet owners to know when health information is reading high or low, prompting them to take necessary measures.
5. Design a smart health device with a lower cost.

6. Provide health tips on how to improve pet health, including hygiene.
7. Implement the ability to track the location of the pet and veterinary hospitals and their health status within a given time range.

1.4 Outcomes expected after completing the health monitoring system

By the end of this project, it is expected that the following results will be accomplished:

1. A well-analyzed and functional prototype will be developed.
2. The user will monitor and visualize pet health through an app.
3. The responsiveness and functionality of the smart health device will be tested.
4. The developed device will be able to communicate with the user and give information on pet health on a given day, and its history will be stored in a database for future reference.
5. The smart health device, including the app, will monitor, analyze, and visualize the pet health information for the owners.
6. The app will be able to track all veterinary hospitals in each locality of the pet owners, including another map for tracking pet's locations.

1.5 Project's Motivations

The idea of owning a pet will continue to exist every century. As time progresses, technology will expand in every aspect, especially in the health sector. The use of traditional methods for monitoring pet health will begin to deplete due to the evolution of artificial intelligence and advanced engineering. My dream and mission are to ensure that this smart health device for pets exists in every part of Ghana or Africa and is ubiquitous and affordable. Nowadays, pet owners have limited time to monitor their pets' health, especially in Africa, and the cost of the devices prevents them from buying one for their pets. Therefore, it can lead pets to shorter lifespans than their original lives, causing much suffering.

1.6 Research Methodology

Below are the research methods used:

1. Literature reviews.
2. Ethnographic research
3. Calculation, simulation, and modeling.

1.7 Scope of the Work

The project focuses on designing and implementing a pet health monitoring system and analysis, incorporating programmable actions and visualization through a mobile app. The scope of work encompasses both hardware and software components. Finally, there will be a GPS tracker for locating pets, maps for locating veterinary hospitals, and pet health tips.

1.8 Facilities used for the research

The places that were involved in the research activity were:

1. Library and internet facilities in Ashesi.
2. Ashesi Fab Lab
3. Ashesi SupportCenter

Chapter 2: Literature Review and Related Works

2.1 Introduction to Pet Health Monitoring and Analysis System

There has been much research on monitoring pet health. It is easier to imagine a life with technological solutions in this modern world. Pet health monitoring systems are part of this technological solution. It became effective during COVID-19 when owners wanted to track their pets' health without visiting a veterinarian. Therefore, there was a high need for it.

Previously, the device was just hardware, but mobile apps and artificial intelligence are now integrated to provide more benefits. This section addresses a variety of related works published on IEEE sites, ScienceDirect, and Google Scholar and provides a critical review of them. Evaluation of the work done was reviewed as well. The two images below show the main categories for pet health monitoring systems.

a

b

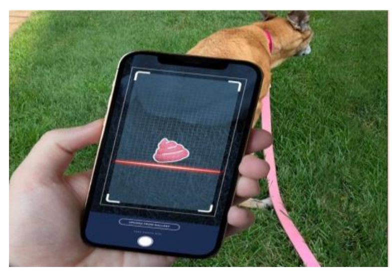


Figure 2.0: (a) Veterinary hospital method. (b) Hardware plus software to monitor pet health

2.1.1 Web Application for Sick Animals Health Monitoring System

The increasing demand for pet health monitoring systems worldwide has made it more difficult for several owners to afford frequent monitoring of their pets by veterinarians. However, this paper seeks to solve some of these problems by designing and developing a web application for detecting pet health using sensors to monitor their health rate and temperature. The paper used hardware to collect the required data and software to provide some needed responses to the users [2]. MySQL was used as the database to store the reading information, which the web will read and write to the database to retrieve some information. MAX30100 was used as the heart sensor to detect the dog's heart rate, and DS18B20 was used as the temperature sensor [2]. The system was to notify the veterinarian when the readings were below baseline [2].

The design, however, comes with some issues. The heart rate sensor consumed more power. The project measured only heart rate and temperature. Additional measures like heart rate variability and activity level will provide more pet health sites. Web applications limit the number of users who prefer mobile apps to access information. According to research, people prefer mobile to web applications for monitoring pet health [5].

a.

b

Figure 2.1: (a) Smart Pet Health collar (b) Flow chart of the web application

Figure 2.2 Web application

2.1.2 Self-powered human-pet interaction system enabled by a triboelectric nanogenerator functionalized pet-leash

Power consumption is one concern of pet health monitoring devices. The paper tends to solve that by comparing its method of generating power to the traditional way, thus using batteries like lithium batteries by utilizing triboelectric nanogenerators to produce electricity for the device [3]. The device was designed using a rotary disk triboelectric nanogenerator functionalized pet leash [3]. It will help reduce costs, optimize device maintenance, and improve uptime by avoiding unplanned outages. The device consisted of embedded TENG, a turntable with a coil spring, and a leash sewn with conductive AG wires [3]. Electricity was generated as the device was pulled out and retracted [3]. The main parameters affecting the device were the electrode's pull-out distance, frequency, and grating degree [3].

The device, however, came with some problems, as it only focused on power management and energy generation; it did not focus on other factors related to the pet's health, like comfortability and others. Additionally, no software was developed to interact with the owner and provide them with some insights about their pet's well-being. Embedding the device with a leash can cause issues with the measurement, despite some benefits like easy movement of the part from one place to another without adding an extra collar to the smart collar.

a

b

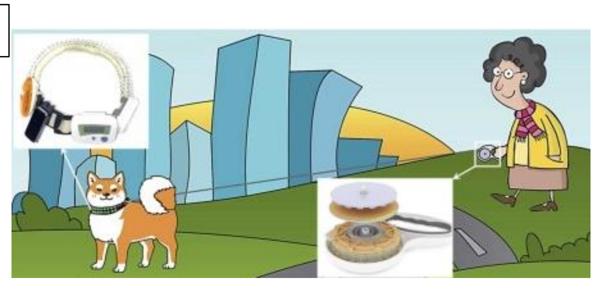


Figure 2.3; (a) Self-powered collar (b) How the self-powered collar works

2.1.3 Smart Pet Clothing for Monitoring of Health and Mood

Pet clothes became popular because of their design. This paper discusses a smart pet cloth that gives pet health insights. The hardware used an analogue-front end circuit to detect electrocardiogram signals and sensors for checking breathing [4]. The software was for health visualization. A Convolutional Neural Network calculates the heart rate (HR), HR variability, and abnormal readings [4]. An iOS app was used to communicate with the hardware device using a cloud server while providing information to veterinarians and pet owners. The microcontroller was a cost-effective, low-power ARM Cortex-M0 MCU, making the design easy to use and suitable for monitoring basic health information. However, it came with some issues as it did not collect data on live pets, which can lead to some uncertainties about actual pet health. Even though the device was designed to be comfortable, it did not account for the difficulty of washing when the cloth gets dirty, as it can affect the sensor performance.

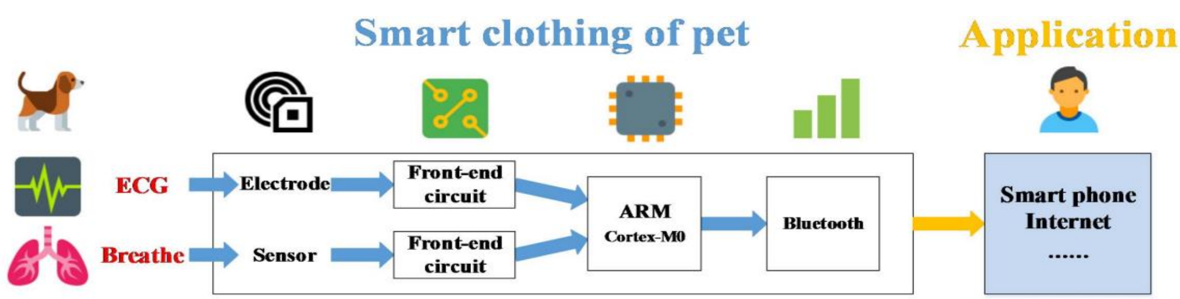


Figure 2.4: How the smart clothing of pets works

Figure 2.5: Smart pet cloth for monitoring their health

2.1.4 Integration of animal tracking and health monitoring systems

Missing pets is one of the pet owners' concerns; there is no means to track pets. The paper discussed integrating GPS and SMS tracking and health monitoring. GPS technology was used to track animals. The system displayed the temperature and location data via SMS based on the user's request with high location accuracy.

The paper successfully developed the system using a temperature sensor (DS18b20) and SMS gateway-based location tracker [7]. Three temperature sensors were used: DS18B20, thermistor (TTCO51022), and DHT11, but DS18B20 was more accurate, thus 99.62%. The system utilizes photoplethysmography (PPG), inertial measurement units (IMU), and NEO 6M SIM modules to remotely and continuously monitor the vital signs of dogs [7]. The GPS and SMS Gateway technology showed an accuracy of 98.3% [7]. This indicates that the system is reliable in monitoring animal health and tracking their location.

However, the paper did not discuss the limitations of GPS accuracy in specific environments (such as dense urban areas) or the potential for signal interference affecting GPS tracking. Additionally, it measured only temperature and the GPS location. No mobile application was developed to make the user visualize how the readings change at a given time. SMS as a communication application is good, but it limits the owner's understanding of how the device works. Finally, the paper still needs to implement the hardware design fully to monitor vital signs and GPS location.

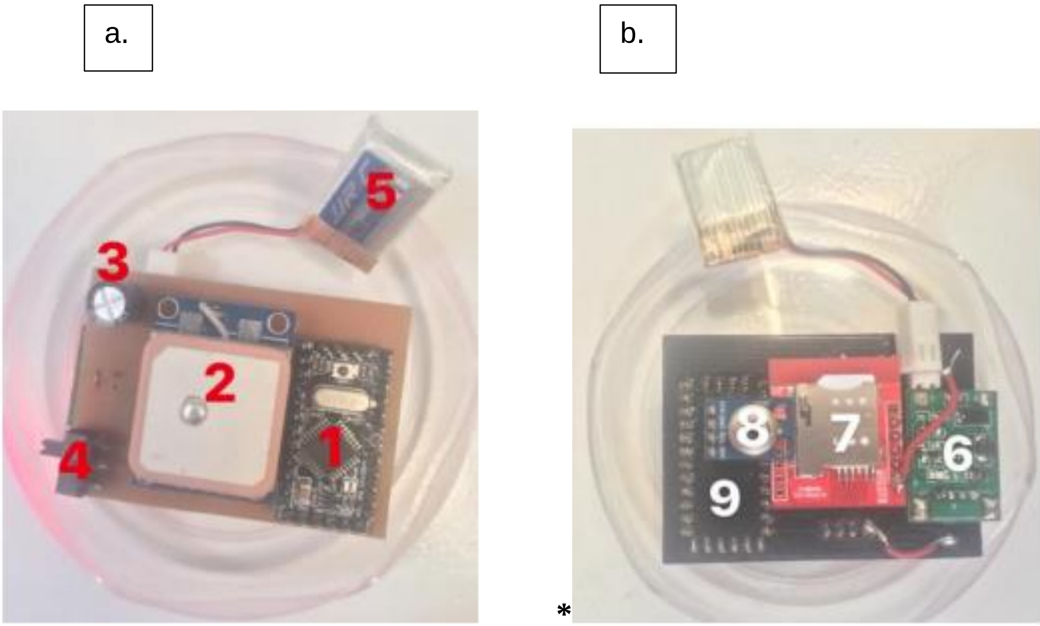


Figure 2.6: (a) Application of modules and sensors (front section) (b)Application of modules and sensors (rear section).

From the images (1) Microcontroller, (2) GSM module, (3) Capacitor, (4) Switch, (5) Lithium battery, (6) Inertial Momentum Unit (7) NEO SIM module, (8) MLX 90614 and (9) DS18B20.

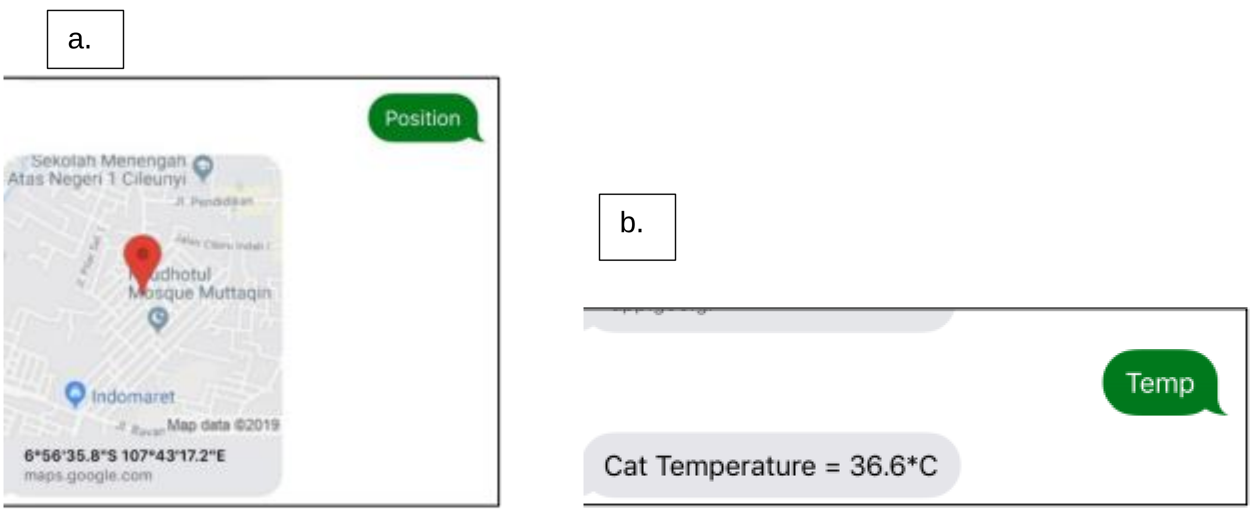


Figure 2.7: (a)Location tracking using SMS. (b) Body temperature of cats via SMS.

2.2 Lesson from the Literature Reviews

The existing works discussed in sections 2.1.1 to 2.1.4 and feedback from the stakeholders provided insights on how to go about the project. First, the initiatives typically used activity levels, temperature, and heart rate sensors. These sensors provided data to support monitoring and diagnostics. Furthermore, every system deployed in [2], [3], and [4] had a central database for maintaining pet information on the Internet. This is crucial for encouraging uniformity and information availability. This implies that the initiative will require an online database that veterinarians and pet owners can access.

Additionally, it is essential to note that [2] and [3] provided a software system to let at least pet owners view the recorded information and provide feedback. The project will adopt this feature to enable owners to view their pet health information and help them locate veterinary hospitals in their locality. The design's simplicity was considered; it is meant to create comfort for pets while monitoring their vital information.

Additionally, cost and power consumption were considered, as the owner wants the devices to last for some days before powering or replacing the batteries. Finally, [4] and [7] thought that the data recorded by the sensor are valid. This ensures the sensors and system's sensitivity, accuracy, and precision. This is especially important for medical equipment, where sensor sensitivity and accuracy cannot be overstated. As a result, the validation of the sensor data will also be considered in this project. The information gathered for this research will be contrasted with data from veterinary sensors nationwide. This will be carried out to ascertain each sensor's sensitivity, accuracy, and margin of error.

Chapter 3: Requirements and Design Specification

3.1 Introduction

The pet health monitoring system comprises sub-systems that will help accomplish the project's objectives. The main requirements for the project will be emphasized, along with how they will influence the design and development of the system. The definition of requirements plays a crucial role in shaping and generating more effective and intuitive solutions for users. The requirements specification in this document will be categorized under the system and user requirements. The practical design and execution of this project will be measured against these system requirements. Subsystems for monitoring and hardware systems make up the main building blocks. Through the mobile app, these owners can manage the device.

3.2 Design Requirements

3.2.1 User Requirements Specification (URS)

They are the features, expectations, and preferences of the stakeholders who will interact with the software system or product. The focus was on "how" to deliver, thus delivery on methodology. Veterinarians and pet owners are the project's primary users. These users will make use of the solution either directly or indirectly. To meet the users' needs, interviews were carried out with pet owners and veterinarians to gather their perspectives about what to expect [refer to Appendix 1 for interview questions]. Furthermore, cues were borrowed from comparable research. To develop a smart pet health device, the URS will outline functional and non-functional safety considerations, environmental issues, and regulatory criteria. Functional and non-functional needs have been separated to aid in managing and organizing the complexities across the development lifecycles. The following section about requirements was based on interviews with people.

3.2.2 Functional Requirements

It specifies features and functions a system must satisfy and fulfill its intended purpose.

1. The device should measure and report key health metrics like temperature, health rate, and activity level.
2. The system should be able to give some health tips for pets.
3. The device should be able to send timely alerts and notifications in cases of abnormal health conditions or irregular behavior.
4. It should be able to collect some information for the veterinarian to make the diagnosis.
5. The device should be able to monitor and view pet health information.

3.2.3 Non-Functional Requirements

It focuses on the attributes, limitations, or features that the system must have rather than simply describing its features. Below is a summary of the non-functional requirements:

1. The device should be user-friendly, safe, and simple to navigate for pet owners with different levels of technical proficiency.
2. The system should be scalable to accommodate additional pets on the user accounts.
3. Ensure data encryption to protect sensitive pet health information.
4. The device should provide accurate health data with a margin of error not exceeding 5%.
5. The device should be cost-effective without sacrificing necessary features
6. The device should operate effectively and consume less power to guarantee a long battery life, which lessens the need for regular recharge.
7. The device should be designed to be non-intrusive and comfortable for the pet to wear.
8. The interface should be interoperable with major web browsers like Chrome, Firefox, and more to guarantee smooth access to the system from various platforms.

3.2.4 System Requirements

It includes the components of the system. This is the setup a system needs for its hardware and software to function efficiently. Since user requirements deal with "what" the system needs to accomplish from the end-user's perspective, the system requirement is meant to delve into "how" to achieve the user requirements. They are more comprehensive because they include technician standards, limitations, design details, and implementation recommendations that direct the development process. Below is a summary of the system requirements.

1. The device must be designed to operate within a specified temperature range of -10°C to 45°C to ensure optimal functionality in various climates.
2. The device should resist dust and water ingress with an IP67 (Ingress Protection) rating, providing durability and protection against environmental elements.
3. The device must be designed to optimize power consumption for extended battery life.
4. The device should support efficient battery charging and indicate when the battery is low, prompting users to recharge the device promptly.
5. The device should provide an intuitive and user-friendly interface for pet owners to view their pet's health data via a mobile application or web portal.
6. The device should implement Robust authentication procedures to guarantee the security of remote interaction, enabling only authorized users to access the smart device.

Figure 3.1: Subsystem of the smart Pet collar

3.2.5 Vital Signs

Pets have measurable vital signs that should always be observed. It shows how their body functions. These vital indicators include heart rate, temperature level, and more. Pets who visit the vet can become more stressed and have higher blood pressure. This is known as the "white coat effect." However, activity level and heart rate variability are crucial to watch. "How can we measure blood flow?" is like how water ripples form in the lake when a rock skips, like how the heart beats and pumps blood in the blood vessels. It is referred to as plethysmography. Essential signals will be considered to emphasize this initiative's vitality for various veterinary monitoring.

3.3 Design Specification

The user's perspective, the pet demand, and the system specification are the three primary areas that this project section will address. The project is split into software and hardware. A discussion of how they will work together is highlighted.

3.3.1 Technical System Design

Table 3.1: Pugh chart for communication technology selection

Criteria	**Bluetooth (IEEE 802.15.1)**	**WIFI (IEEE 802 11B)**	**Zigbee (IEEE 802.15.4)**	**LoRa**
Range	0	1	2	3
Data Rate	2	3	0	-1
Cost	2	2	1	0
Availability	2	2	-1	0
Power consumption	2	1	3	1
Battery life	2	1	2	1
Security	1	1	3	1
Reliability	2	2	2	1

TOTAL	**13**	**13**	**12**	**6**

Table Pugh chart for sensor communication protocol selection

Studying communication publications served as the inspiration for the data. The parameters used to select the microcontroller include memory, architecture, performance, price, power usage, and direct Bluetooth and Wi-Fi connectivity.

Explanation of the Pugh chart

LoRa has a long-range (up to 10 km in rural places), followed by Zigbee, which has a range of 100m but can extend with mesh networking; Wi-Fi has a range of (up to 100m), and Bluetooth has 100m. Bluetooth has a data rate of 3Mbps, and Wi-Fi has 11Mbps, Zigbee has up to 250bps, and LoRa has up to 50bps

Table 3.2: Pugh Matrix for microcontroller selection

The higher the number, the better that component is for that section.

Criteria	**Esp32 chip (Baseline)**	**Arduino Uno**	**Arduino Nano RP2040 chip**	**Esp866 chip**	**Raspberry Pi**	**Arduino Nano**
Cost	1	2	0	2	-2	3
Speed	2	-2	1	2	3	-3
WIFI	3	-3	3	1	0	-2
Bluetooth	3	-5	3	-3	2	-4
Power Consumption	1	1	3	-1	-3	2
Architecture	1	-2	2	0	3	-1
Size	3	-1	2	1	-2	-3
TOTAL	**13**	**-10**	**14**	**2**	**-2**	**-8**

Table: Pugh chart for microcontroller selection

Table 3.2 shows that the **Arduino Nano RP2040 Connect** has the highest total weight, making it the most suitable microcontroller for the project. This is because the smart collar should be small, have a Wi-Fi connection, and have low power consumption.

Even though the Arduino Nano RP2040 was the top choice, it costs $25, but the cost is the topmost priority; therefore, the second best, the ESP32-S2 WROOM chip, was used. The design of the schematics and simulation of the primary choice were tested to ensure the design worked. However, the ESP32 chip was used to build the final prototype.

3.3.2 Hardware Components

This part consists of different types of sensors, microcontrollers, and power systems. The device will contain some sensors that will capture the vitals and transmit them to an online database, which will then be used for backend work and front-end display of the vital information. The hardware will contain sensors like temperature sensors, pulse rate sensors, and accelerometer sensors. Figure 3.2 shows all the relevant parts of the hardware.

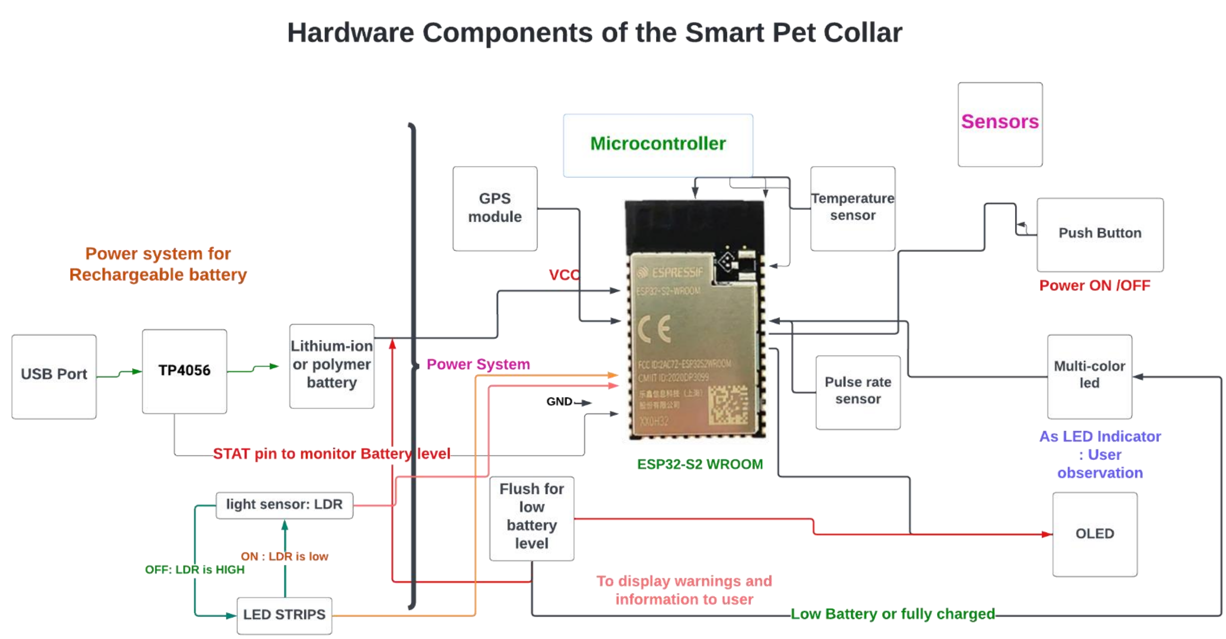


Fig. 3.2: Block diagram of electronics design

3.3.3 Contextual Diagram

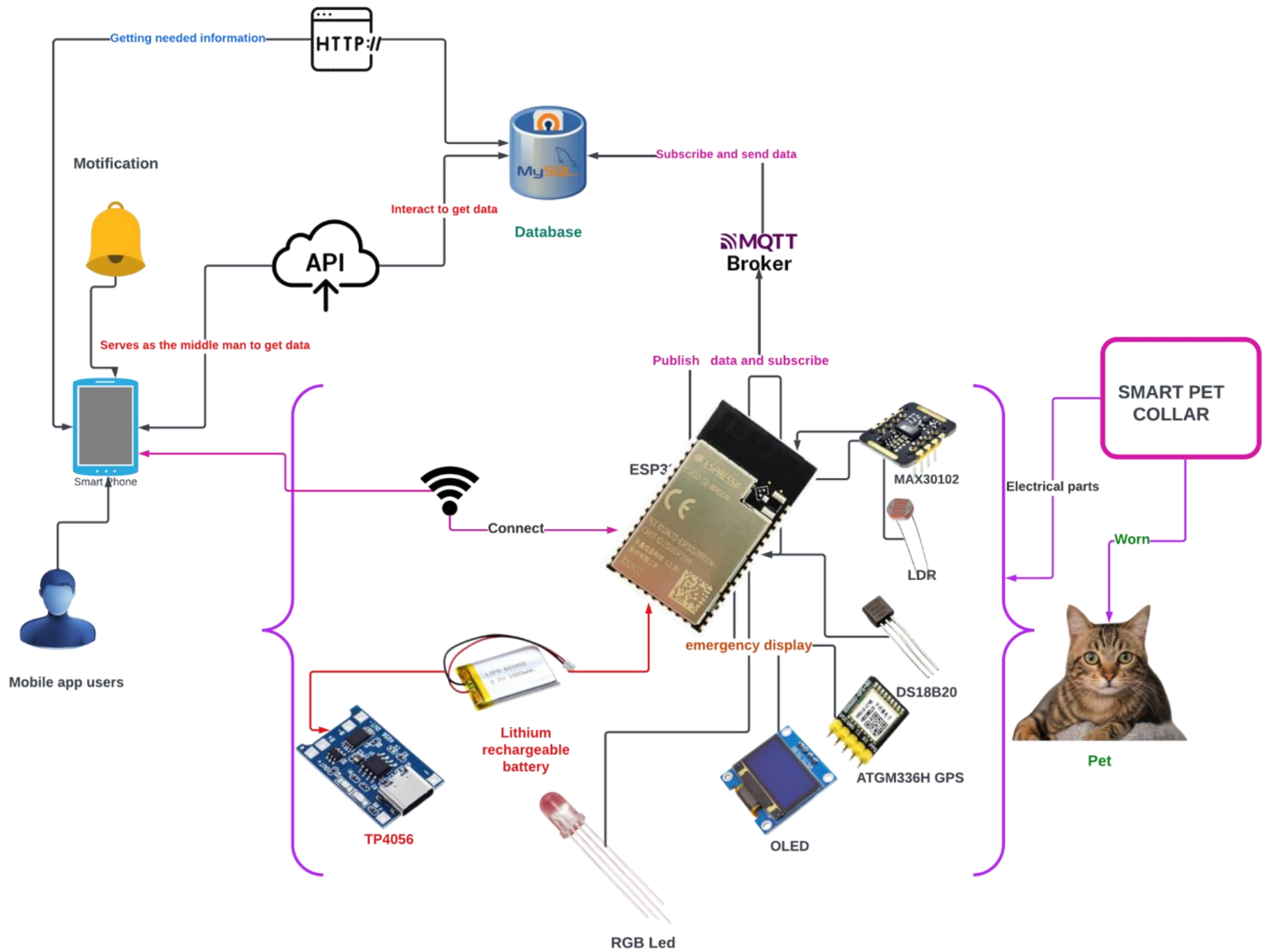


Figure 3.3: Contextual diagram for Smart pet collar

3.3.4 Selection of Hardware Storage –Smart Pet Collar Versus Smart Pet Health Container

Two designs were the most effective for the smart pet health monitoring system: the smart pet collar and the smart pet health container. Based on feedback from stakeholders and technical feasibility, the following comparison shows each design's key differences.

Table 3.3 Comparison between smart pet collar and smart pet container

Features	Smart Pet Collar	Smart pet health container

Portability	Highly portable.	Not portable
Comfort of pet	Comfortable for pets; lightweight	It may not be comfortable; stationary
Space efficiency	Occupies minimal space.	Occupies more space
Continuous monitoring	Monitors vitals continuously.	Only monitors vitals when the pet is inside.
Owner preference	Pet owners prefer it for its portability.	Less preferred due to size.

Table 3.3 shows that a smart pet collar is preferred for its continuous monitoring abilities, portability, and convenience for pet owners. This design meets the primary requirement of being lightweight, comfortable for pets, and space efficient.

3.4 Software Components

The interface gives pet owners access to health data about their pet(s), offers advice and tips, and assists users in finding nearby veterinary hospitals in an emergency. The software comprises a mobile application and an internet-based database. The integration of the two subcomponents to communicate with the hardware is illustrated in Figure 3.3. The hardware will directly send the pet's essential data into the database via a simplex communication mode. The mobile application can then retrieve and modify the data in the database.

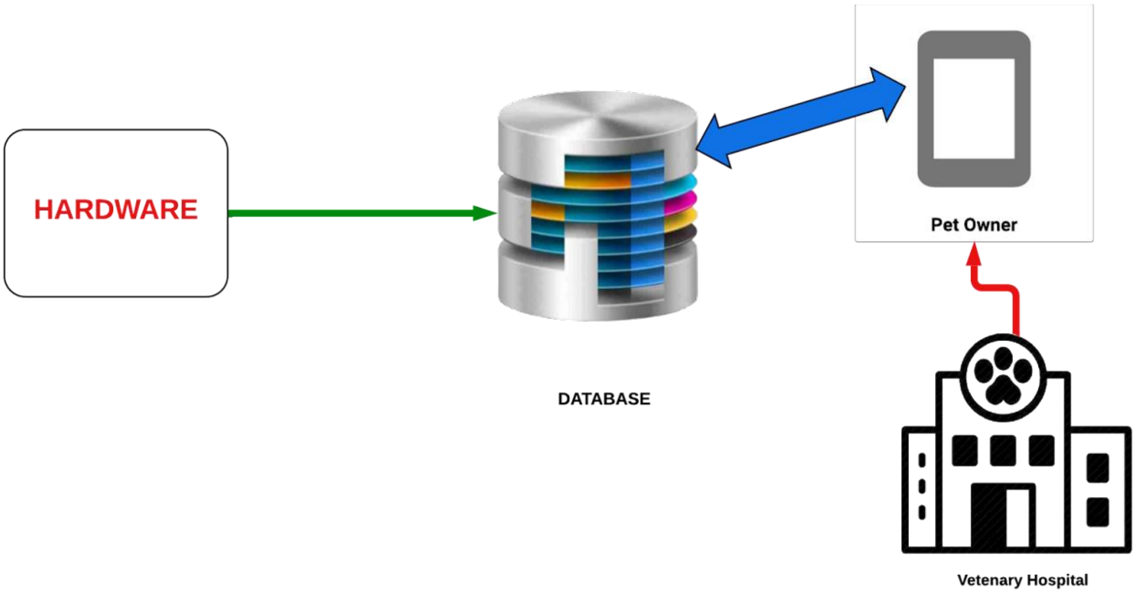


Figure. 3.4: Hardware and Software Integration

3.4.1 Application Layer Protocol

Figure 3.3 shows how the hardware and software interact together. The hardware must communicate with the cloud platform to exchange data and receive commands. The application layer protocol handles different communication protocols like **MQTT (M**essaging **Q**uery **T**elemetry **T**ransport), **HTTP (H**ypertext **T**ransfer **P**rotocol), **CoAP** (**C**onstrained **A**pplication **P**rotocol**),** and **XMPP** (E**x**tensible **M**essaging **P**resence **P**rotocol) and manages interaction with APIs. All these application layer protocols are founded on **TCP** (Transport Control Protocol) and **UDP** (**U**ser **D**atagram **P**rotocol).

The protocol ensures that the device is compatible and interoperable with other devices it communicates with. The protocol must provide well-secured information. Therefore, a Pugh chart was done on the most prominent application layer protocol for Internet of Things projects.

Table 3.4 is used to select the application layer protocols. The perfect one has higher marks, and the less effective one has a lower mark. **3:** Best; **2:** Better; **1**: Good.

Table 3.4: Pugh Chart for selecting Application Layer Protocol

Criteria	**Baseline (HTTP)**	**MQTT**	**CoAP**
Efficiency	1	3	3
Reliability	1	3	2
Low Bandwidth	1	3	2
Low Latency	1	3	2
Security	2	3	2
Real-time communication	1	3	2
Scalability	1	3	2
Interoperability	2	3	1
Cost	2	2	2
TOTAL	**12**	**26**	**18**

The criteria listed are the main things being considered for the project. As the device must send information faster without too much delay, the cost of delivering the information should be reduced to reduce the power consumption it takes to send information to the database. Since we are dealing with health information, security is a priority; therefore, TCP over UDP. TCP also takes note of reliability and efficiency. From the Pugh chart, it was determined that MQTT is the most suitable for the project based on the total value of **26** as it has the crucial parts of the project criteria.

3.4.2 Database

The RDBMS database type is being used for this project. Since data is stored in tabular format, Oracle, MySQL, and MYSSQL are a few examples. MSQL for testing and Firebase will be used as the databases for storing the pet information on the pet's owner and sensor readings.

3.4.3 Mobile/Web application

This section is divided into front-end and back-end development. The front-end part will provide a platform for health visualization. It will provide an interface for the pet owner to view the pet's vitals. Additional information will be added to the interface, as it will give a part that shows the measures pet owners can take to improve their pet's health. It will allow pet owners to register more than one pet and view their vital information and location independently.

The front-end interface will be developed using Flutter, JavaScript, and Ajax. A responsive and user-friendly UI will be the ultimate part of the mobile app; therefore, Firebase will be adopted to achieve that. Django will be used for the backend part since it is a web framework for Python. It will serve as the middleman for the front end and the database by retrieving, adding, or deleting data from the database. Django will also contain some backend calculations to show information about the vital signs while the hardware reads information from the outside world.

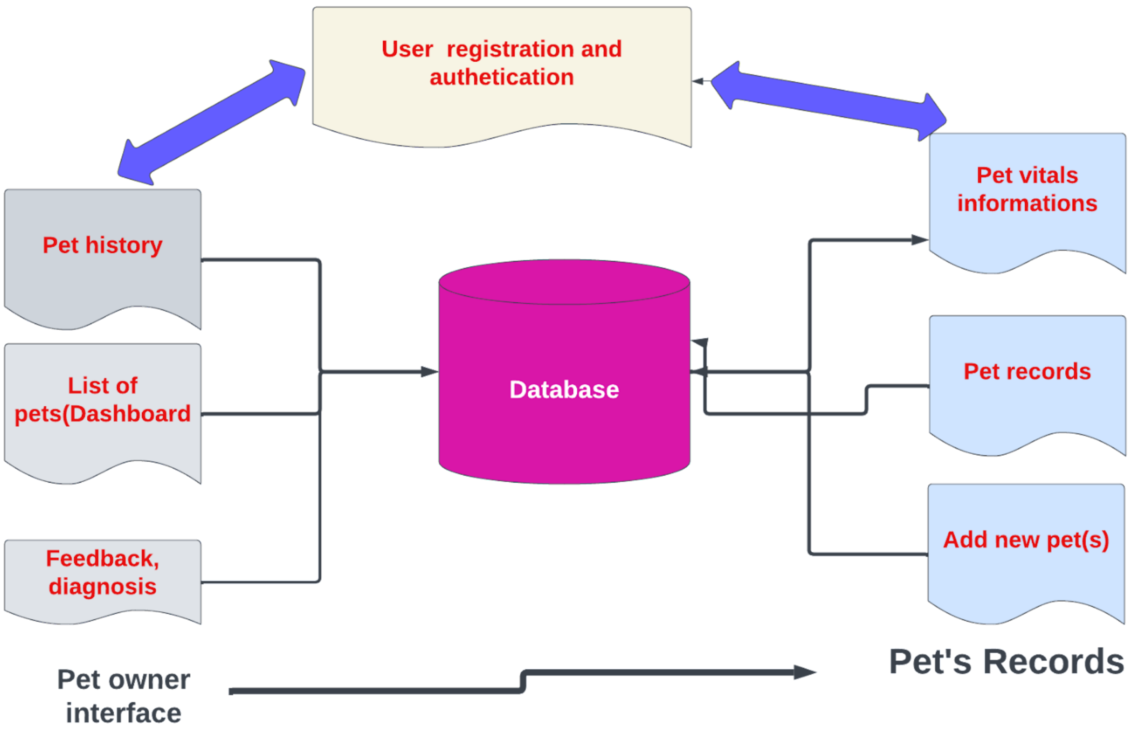


Figure 3.5: User interface framework for the mobile application

3.5 Electronic Components

This part of the project discussed the various devices, sensors, and components used in building the smart pet collar. The respective components are carefully selected by evaluating passive and significant elements for workability. Below is a list of the components.

3.5.1 ESP32-S2-WROOM chip

Tensilica Xtensa dual-core 32-bit LX6 microprocessor ESP32-S2 chip, a potent, generic WIFI+BLE+MCU module that targets a variety of applications ranging from low-power sensor networks to the most demanding jobs, is contained on the development board. The ESP32-S2 chip is crucial to this module [7]. The integrated chip is designed to be scalable and adaptable. The CPU clock frequency can be adjusted from 80 to 240 MHz, and two CPU cores can be regulated independently. [7].

3.5.2 Light Dependent Resistor (LDR)

Other names include photoresistor, photocell, and photoconductor. A particular kind of resistor called an LDR varies in resistance in response to the quantity of light that strikes its surface [8]. The resistance varies when it is exposed to light. LDRs have many uses, but for this project, they will either turn on the LED strips at night or turn them off when it's light outside. LDR has a resistance of one MOhm in the dark and a few KOhm in the bright.

3.5.3 Organic Light-Emitting Diode Display (OLED)

An organic layer in an OLED reacts to an electrical current by emitting light. Like traditional diodes and LEDs, it produces electrons and holes utilizing organized molecules rather than layers of n-type and p-type semiconductors. It works by using organic molecules' electroluminescent qualities. It comprises essential elements such as cathodes, emissive and conductive layers, and anodes. Due to their emissive nature and higher efficiency than LCDs, they do not require a backlight [9]. Because it typically runs at a voltage of 3.3V and 0.06 W, it consumes little power.

3.5.4 MAX30105 - Pulse rate and Heart rate sensor

The heart sensor is the biosensor module that measures heart rate and pulse oximetry. Internal LEDs, optical components, photodetectors, low-noise circuitry, and ambient light rejection are all included. It runs on two different power supplies: one for 1.8V and one for 3.3V, which powers the internal LEDs.

3.5.5 DS18B20 - Temperature sensor

It is an integrated digital temperature sensor produced by Maxim. It is a single-bus addressable temperature sensor that provides 9-bit to 12-bit Celsius temperature readings. It operates from 3.0V to 5.5V supply voltage and consumes shallow current (less than 150mA).

3.5.6 MPU6050 module - Accelerometer

The MPU6050 is the primary component, as indicated in Appendix 3. A voltage regulator is further needed because the module operates on 3.3V. A 4.7k resistor is used to draw the IIC lines high, and another 4.7k resistor is used to bring the interrupt pin down.

Through the IIC bus, we may read data thanks to the MPU6050 module. The mechanical system will adapt to any change in motion, changing the voltage. The IC then employs its 16-bit ADC to precisely detect these voltage variations, stores the data in the FIFO buffer and raises the INT (interrupt) signal.

3.6 Circuit Diagram

Using Electronic Design Automation, the circuit schematic for the pet health monitoring system was created using EasyEDA and Autodesk Fusion 360. The different relationships between the elements mentioned in Section 3.2.4 are depicted in Figure 3.7. Additionally, crucial passive parts were incorporated for effectiveness and safety—for instance, diodes, capacitors, and resistors.

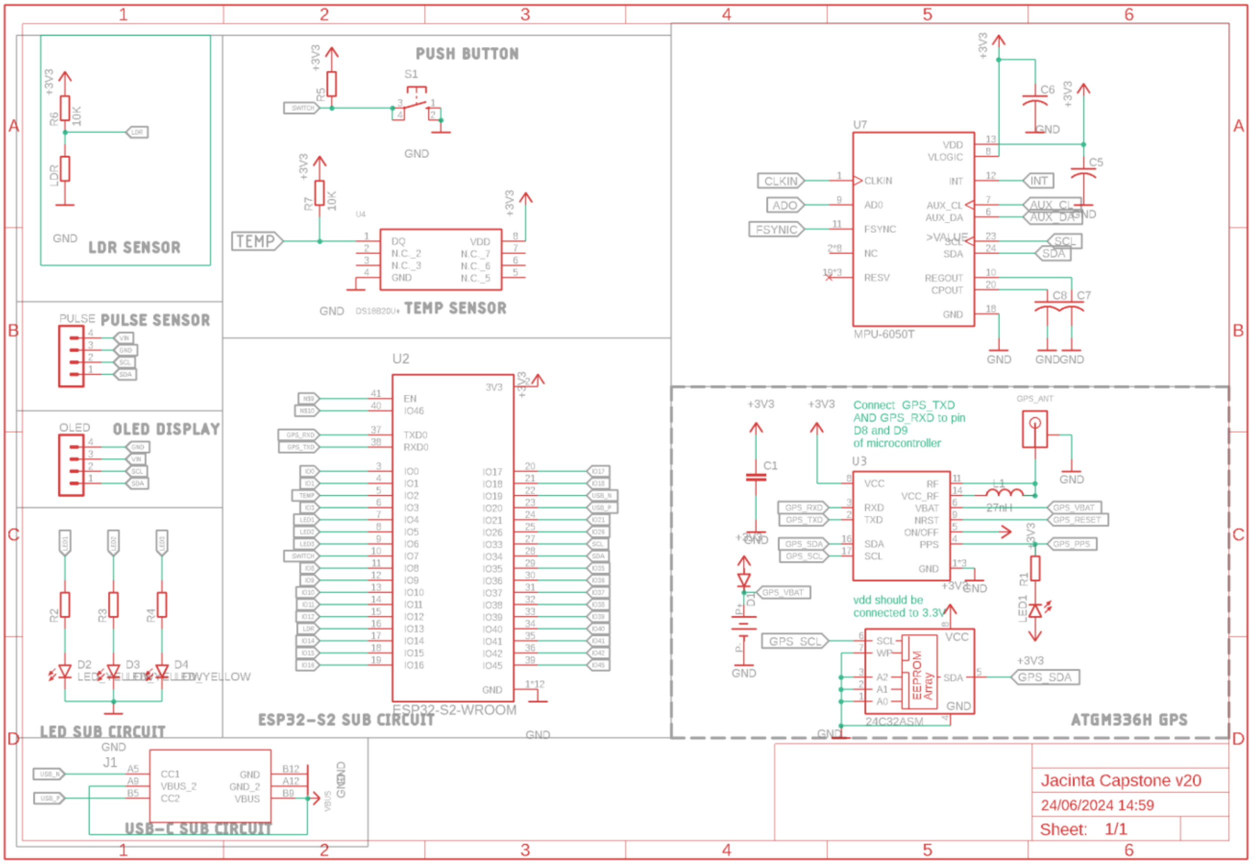


Figure 3.6: Schematics of the intelligent collar hardware part with the modules

3.6.1 Explanation of Schematics – ESP32 -S2 WROOM

In this project, the ESP32-S2 microcontroller is the central hub, coordinating data collection and control for various components. The MAX30100 Pulse Oximeter module measures the pet's heart rate and SpO2 levels by using infrared and red LEDs to detect blood oxygenation and pulse rate, communicating this data to the ESP32-S2 via the I2C interface. The MPU6050 module, including a 3-axis accelerometer and a 3-axis gyroscope, provides data on the pet's motion and orientation, interfacing with the microcontroller via I2C. An LDR (Light Dependent Resistor) detects ambient light levels, enabling adjustments to the OLED display brightness or triggering actions based on lighting conditions. This sensor is connected to an analog input pin on the ESP32-S2. For temperature monitoring, the DS18B20 temperature sensor was used to communicate with the microcontroller through the One-Wire protocol to provide accurate body temperature readings of the pet. The OLED display, connected via I2C, shows real-time data, including heart rate, SpO2 levels, activity status, temperature, date, and time. An LED indicator is incorporated to alert users to abnormal readings; it is connected to a digital output pin on the ESP32-S2 and lights up when any sensor values are outside expected ranges.

The system is powered by a LiPo rechargeable battery, managed by a voltage regulator to ensure a stable voltage supply to all components. A USB-C port is included for convenient battery charging, with the battery management system overseeing charging, discharging, and protection mechanisms. All components work together under the control of the ESP32-S2 to provide comprehensive monitoring and alert functionalities for pet health.

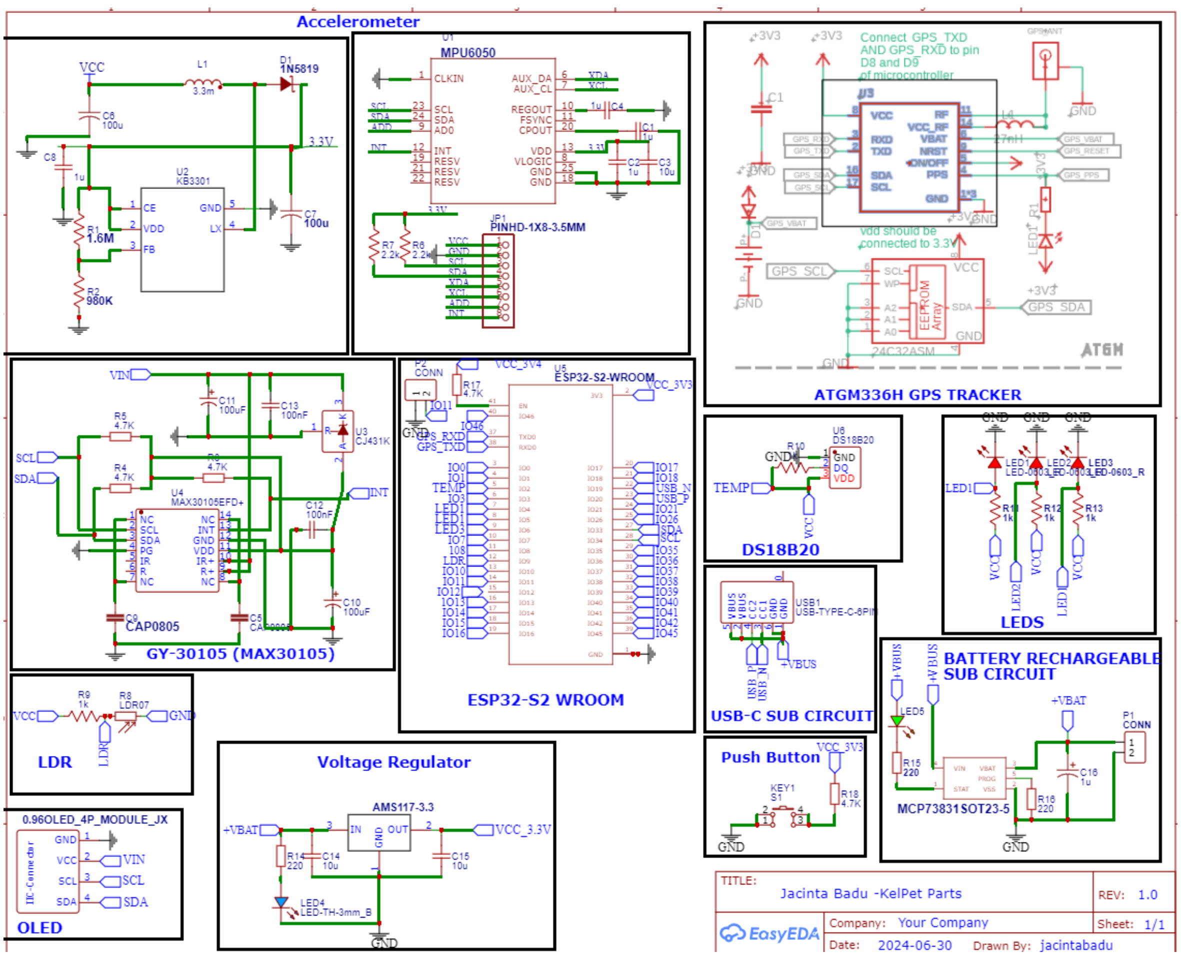


Figure 3.7: Schematics of the smart collar parts with the module's schematics expansion

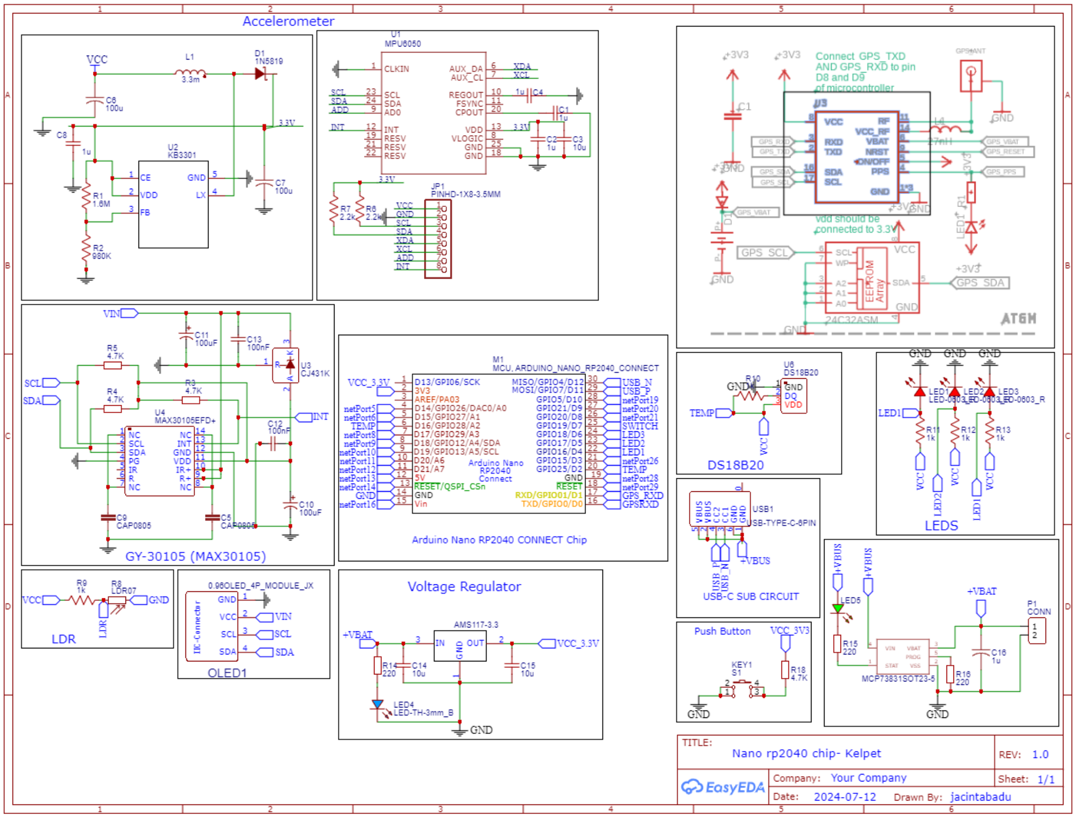


Figure 3.8: Electrical schematics of electrical parts with Arduino nano rp2040 chip

Chapter 4: Methodology- Important calculations and considerations

4.1 Introduction

Calculations were needed for the circuit and program to run efficiently. Table 4.1 lists components, prices, quantity, and voltage ratings. The information was obtained from their respective datasheets and online purchases.

Table 4.1: Components used and their prices

Number	Electronic name	Electrical term	Amount used	Ratings	Supplier	Cost (Cedis)
1.	OLED 0.96-inch 128x64 I2C	OLED	1	DC 3.3 ~ 5V, 20.7mA	Daakye electronics	59
2	Lipo Battery module	TP4056 Li-ion Battery Charging Module	1		Ashesi	16
3.	Dallas temperature sensor	On-board DS18B20	1	DC 3.3V ~5V	Daakye electronics	6
4.	USB-Type C		1		Nauvitel Electronics	20
5.	Heart rate sensor	Max30100	1	DC 3.3V ~5V	Nauvitel electronics	44
6.	GPS module	ATGM336H	1	DC 3.3V ~5V	Ashesi Fab Lab	65
7.	Accelerometer	MPU6050	1	DC 3.3V ~5V	Ashesi Fab Lab	25

Overall total price = **300 cedis (including PCB board price)**

4.2 Calculations for Current and Power

4.2.1 Using Calculation to choose the suitable power source

Table 4.2: Component ratings summary

Electrical parts	**Current (A)**	**Voltage (V)**
OLED	$3 * 10^{-3}$	3.30
Esp32-S2 WROOM	$340 * 10^{-3}$	3.30
MAX30105 (measurement mode)	$60 * 10^{-6}$	3.30
MA30105 (Shut down mode)	$6 * 10^{-6}$	3.30
Mounted SMD	$60 \ * 10^{-3}$	3.30
DS18B20	$3 * 10^{-3}$	3.30
MPU6050 accelerometer	$3 * 10^{-3}$	3.30
ATGM336H GPS	$25 * 10^{-3}$	3.30
Total	**Current: 0.37706**	

Based on the table above, the total power and current used by the electronic components, thus the maximum power is calculated as;

From Ohm's law: $V = IR$, I = current, R= Resistance … (1)

$P_T = IV$, $P_T = \ Total\ Power$ … (2)

$P_T =$ [(3 $* \ 10^{-3} * 3.30$) +($600 * 10^{-6}$*3.30) + (340 $* \ 10^{-3}$*5) +2(60$*$ $10^{-3} * 3.30$) +

(3$*$ 10^{-3}*3.30) *3 + (25$*$ 10^{-3}*3.30)] = **1.630398W approximately 1.63W**

Therefore, the power source to be used should be able to provide 377**mA-** approximately 380mA.

The Lithium Polymer battery used has a voltage of 3.7V and power of 7.4W.

$P = \mathbf{7.4W}(> \textit{the total re}$quired circuit power: 1.63W)

$$I = \frac{P}{V} = \frac{7.4}{3.7} = \mathbf{2A}\ (>\ \textit{the total current consumption of the circuit};\ \mathbf{377mA})$$

4.2.1 Calculation for the Selection of Lithium Polymer Battery

The device will work for 5 hours; thus, the battery's capacity can be calculated as follows:

$Capacity, C = Current\ (Amperes) * Time\ (Hours)$, unit; Ampere Hour, mAh

$$C = \mathbf{377 * 5 = 1885mAh}$$

Therefore, the battery used had a 3.7V capacity of 2000mAh > 1885mAh **[**check Appendix 3**]**.

4.3 I^2C (Inter-Integrated Circuit) Consideration

There is only one I^2C Interface in the chip. GPIO21 for SDA and SCL is 22. Initially, it fitted the project well; however, with the addition of MAX30105, an issue arose; therefore, an extra with the help of research. Below is the code.

```
#include <Wire.h>
#define SDA 21
#define SCL 22
Wire.begin(SDA,SCL);
```

4.4 Writing Efficient Code:

Code can be efficient if written well. To prevent that, specific protocols were used to improve the code.

1. ESP32's watchdog timer reset the board when the WiFi connection, MAX30105, and GPS module failed. As there can be an unreliable moment, resetting must be done to make the board work again. The code of the watchdog timer is.

   ```
   esp_task_wdt_reset();
   ```
2. Functions were created to organize the code well.

3. Code separation into source codes and header codes (#include .c and .h): It was a proficient way to structure my code to be easily read and organized. An example was separating my Wi-Fi bitmap from other codes. Therefore, it was placed in a .h file.

4.5 Machine learning model formulas used

For the logistic regression, the formula used for the training was.

$$f = \frac{1}{1 + e^{-z}}, Z = y = linear\ function = b_1x_1 + b_2x_2 \ldots$$

Where y is the dependent function, and x is the independent function.

Chapter 5: Implementation

5.1 Introduction

Based on the design decisions covered in the preceding chapters, this chapter will outline the procedures followed to construct, design, and execute the project. The 6.15 cm x 4.39 cm PCB board design, a software design flowchart, machine learning models for assessing the pet's general health state, the hardware code implementation, and the overall integration in detail. We will examine every task completed during the implementation of the Kelpet project.

5.1.1 Design of the PCB Board

A printed circuit board, or PCB, was used to implement the hardware part's circuit design. This was done to make the device visually appealing, easy to operate, and manageable. The top section of the finished 6.15cm x 4.38cm PCB is shown in Figure 5.2, while Figure 5.1 displays the routing layout both with and without the ground copperplate.

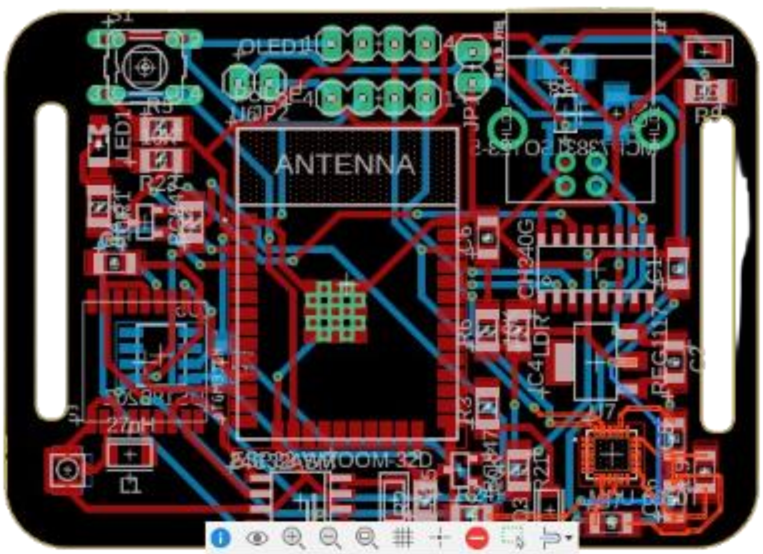


Figure 5.1: PCD Design of the Smart Pet Collar (Kelpet components)

5.1.2 Smart Pet Collar Design

A critical part of the project is the physical part placed on the pet collar. The collar has space for the OLED to display its battery level. The inner part of the collar will house the electrical components, including the sensors. Below is the 3D model and SolidWorks design.

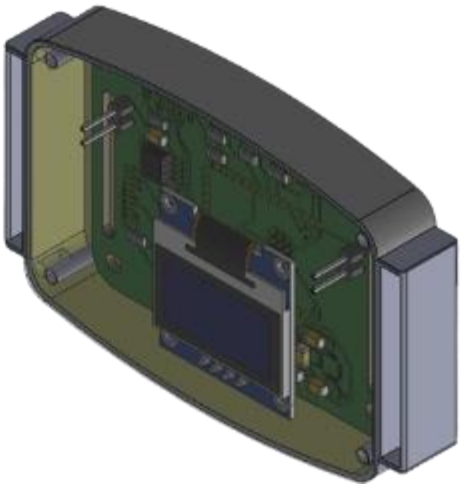


Fig 5.2: 3D model of the intelligent pet collar housing the electrical parts

5.1.3 Complete Hardware System Setup

After individually testing the hardware components, all components were arranged onto the 6.15cm x 4.38cm PCB. The lead paste was applied to the PCB for the components and placed on a heating machine to harden the board. Using a soldering gun, a lead wire was used to solder the heart sensor and temperature sensor to the back part of the PCB. The ESP32-S2 was soldered first on the board, and after that, the sensors were soldered on the board, followed by the USB-type C, including the resistors and the capacitors. The push buttons were placed. Next, the MAX30105 and the OLED were soldered onto the board, which covered the ESP32-S2 chip. Further testing on the design was done to ensure that all the parts function effectively and meet the proposed system specifications. Find the complete setup in Figure 5.3 below.

Figure 5.3: Prototype Circuit

5.1.4 Hardware Component Backend Code

A program was written to manage and control the device to ensure the hardware works as expected effectively. For the Arduino Nano RP2040, code was written to test its workability; it had a Wi-Fi connection, "**#include WiFiNiNa.h**," which helped it to connect to the Wi-Fi and even use it to connect to the MQTT, and data from the DS18B20, and GPS data were published to the subscribed topics on "HiveMQTT.com." For the MAX30105, the sensor must be stopped after initializing [a] and start again when the Wi-Fi and the MQTT connect to prevent it from not recording values [b] when the code runs. The code implementation can be seen in Figure 5.4.

a

```
initPulseOximeter();
pox.setOnBeatDetectedCallback(onBeatDetected);
stopPox();  // Stop the Pulse Oximeter
```

b

```
void loop() {
  if (WiFi.status() == WL_CONNECTED && client.connected()) {
    if (!poxTaskStarted) {
      startReadPoxTask();
      poxTaskStarted = true;
    }
  } else {
    reconnect();
  }
```

Figure 5.4 Implementation of MAX30105

The Heart Rate Variability was calculated using the IR readings from the MAX301xx library to calculate the HRV. The regular beating of the heart is called "sinus rhythm."

This information was sent to the database with the help of the Django backend code, which was done by connecting the MQTT broker to it to send data. "**#include WiFi.h**" was used to connect to the WiFi. SmartConfig was used to remotely change the Wi-Fi connection remotely, preventing the "hard coding" of WiFi credentials, as in Figure 5.5. Apart from SmartConfig, one can use WiFiManager to change the WiFi credentials automatically, but this requires the IP address of the Esp32. The SmartConfig connects the WiFi on the phone to the esp32. However, one needs to download this app, "EspTouch," from the **Play Store**, enter the password for your phone's WiFi connection, and select "multicast" and "Confirm." After this process, the board will connect to the ESP32-S2. Initialization of the sensor's libraries was done in the setup code, and the sensor's values were recorded in the loop function. However, the sensors were initialized before connecting to the WiFi and MQTT.

```
WiFi.mode(WIFI_AP_STA);
WiFi.beginSmartConfig();

// waiting for SmartConfig Packet from Mobile
Serial.println("Waiting for SmartConfig.");
while (!WiFi.smartConfigDone()) {
  delay(500);
  Serial.print(".");
}

Serial.println("");
Serial.println("SmartConfig received.");
```

Figure 5.5: Smart Config implementation

5.1.4.1 Sensory Data

The heart rate and oxygen saturation, SpO2, sensor used is MAX30105, thus measuring the amount of oxygen-carrying hemoglobin in the blood. For activity level, the MPU6050 chip, a low-cost sensor, measured the orientation of the pet's body. It gave three axes, thus the roll, pitch, and yaw. DS18B20 was used to measure the pet's temperature by measuring the forward voltage drop across the diode, and the silicon temperature was then calculated.

5.2 Result of Software Implementation

These innovations—such as hardware programming code, online databases, and mobile applications—form the foundation of the software. This study section offers a detailed analysis of the different software features and implementation designs.

5.2.1 Database Management

We work with people and their pets. Thus, it is necessary to save their data on a safe and accessible platform. As a result, the user, pet, and system data were managed via an internet database. When designing the database, users, sensors, health data, and the necessary system information were considered. A table was created for the heart rate sensor readings (MAX30105), another table for temperature readings (DS18B20), GPS tracker, and accelerometer readings (MPU6050). These sensor readings were linked to the collar table. In the user table, the collar ID tends to be a foreign key to individualize a specific pet collar to a particular pet and owner based on the collar's serial number.

#	Name	Type
1	**TempId**	int(11)
2	**time**	datetime(6)
3	**temperature**	double
4	**collar_id**	int(11)

#	Name	Type
1	**HeartRateId**	int(11)
2	**time**	datetime(6)
3	**HeartRate**	double
4	**collar_id**	int(11)
5	**SPO2**	double

#	Name	Type
1	**GPSId**	int(11)
2	**GPSTime**	datetime(6)
3	**longitude**	decimal(9,6)
4	**latitude**	decimal(9,6)
5	**altitude**	double
6	**speed**	double
7	**collar_id**	int(11)

#	Name	Type
1	ActivityId	int(11)
2	time	datetime(6)
3	activity_x	double
4	activity_y	double
5	activity_z	double
6	collar_id	int(11)

Figure 5.6: Database table of the sensor readings

5.2.2 Flow Charts

An overview of the different choices and actions made by the hardware device is provided in Figure 5.7. The system navigation for seeing information and receiving warnings was demonstrated. The smart collar would record temperature, activity level, heart rate, and GPS tracker information using a microcontroller with sensors, passive components, and a battery attached. This data will be sent to a database using MQTT and WiFi. The backend (**Django Backend**) will retrieve the data from the database, and this information will be analyzed using probabilities and "IF ELSE "statements to know the pet's temperature status and heart rate. Machine learning was done to predict whether a pet was active, mini-active, or inactive using three models, and the best accuracy was used for the prediction. The overall health status was also determined, such as healthy, unhealthy, or not entirely healthy, as this was done using the machine learning classification method to classify the health status.

For the front end (**FLUTTER**), there were graphs (visualization of the health vital signs), registration for both user and pet, health tips, and a location tracker (veterinary hospital and pet location). In case there were abnormalities in the readings, like inactive or unhealthy, a notification will be sent to the user, and health tips will be shared to guide them on some things they can do before visiting the veterinarian to save them from danger. The LEDs on the device, too, will show a warning in case the user is not close to their phone to keep track of the pet's health. Green means the pet is healthy, red means the pet is unhealthy, and yellow represents the tendency to be sick. A dashboard has been provided for users to check their pet's health status. Still, a user can only add a pet if they have another smart pet collar by entering the device's serial number before they can register. Every pet's information, including the location, is tracked separately. Finally, users can still log in to view their pet's health or log out of the system.

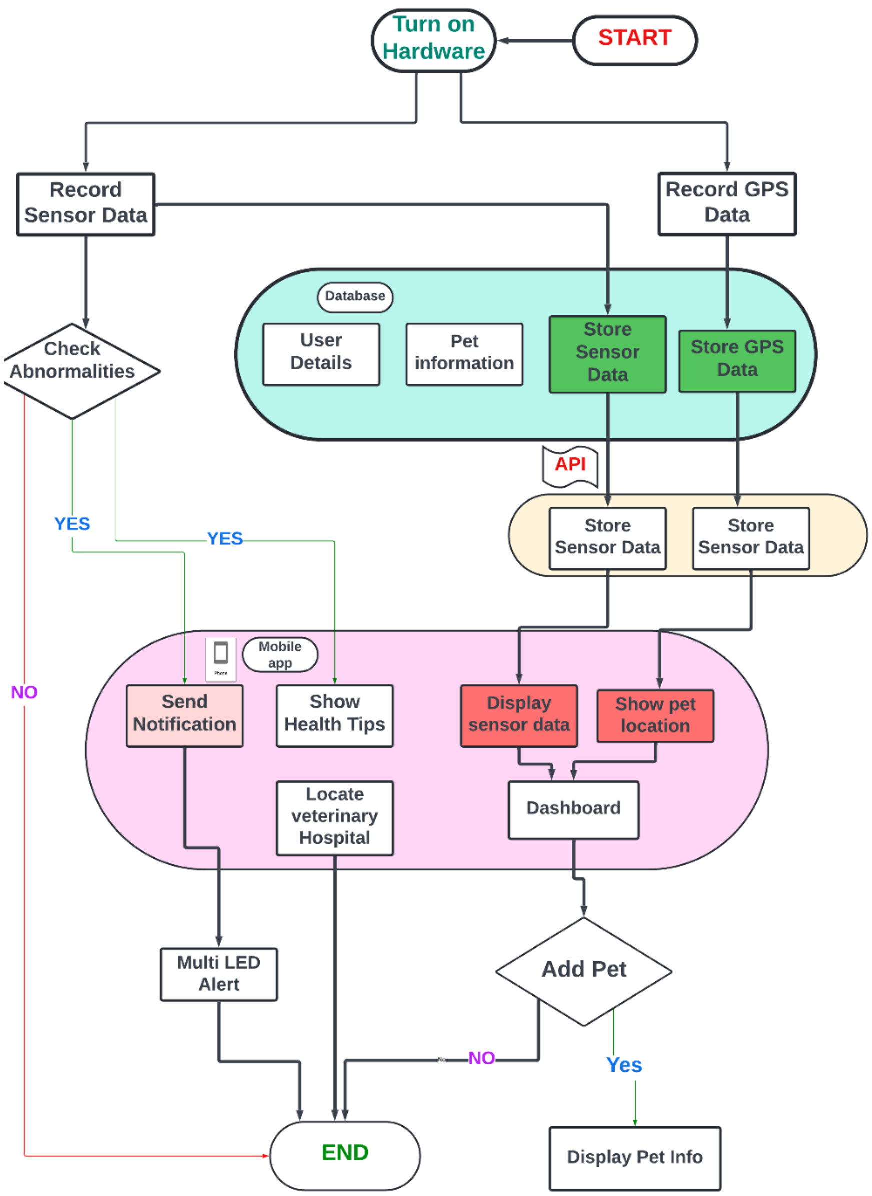


Figure 5.7: Software flow chart of how the software and hardware behave in coding.

5.2.3Machine Learning

Prediction models help healthcare professionals to make clinical decisions. An accurate prediction aims to provide pet owners with risk stratification to support tailored clinical decision-making to improve pets' outcomes. Hyperparameter tuning was used to find the optimal combination of hyperparameters that yields the best performance for machine learning [11].

5.2.3.1 Goal of the model

Determining the general objectives of the model is the most crucial phase in creating a prediction model. In addition to guiding the identification of relevant source data, predictor selection, and model construction, carefully selecting the portion of the population you want and the outcome and how it is determined also defines the generalizability of the finished product. Seven primary stages were taken to create health status and activity level prediction models.

5.2.3.1.1 Seven steps for developing valid prediction models

a. Determining the prediction problem
b. Code predictors: Statistics were used to find them.
c. Specify the model.
d. Estimate model parameters.
e. Model evaluation
f. Model validation
g. Presentation of the model using Seaborn or Pyvis.

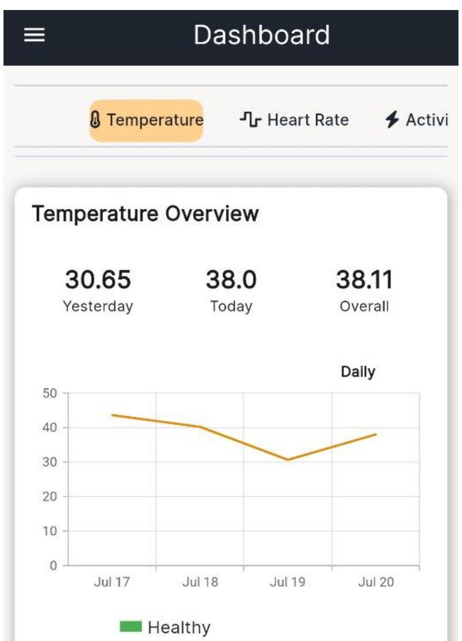
5.2.3.2 Overall health status

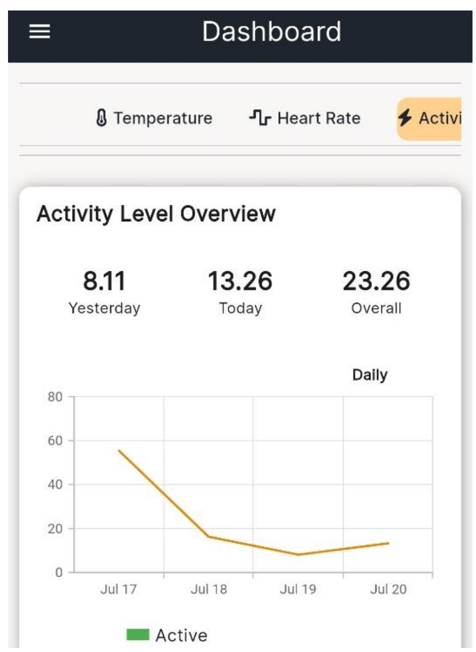
The training was done on the obtained dataset [refer to Appendix 5]to determine whether a pet is healthy based on vital signs sensors. Machine learning is divided into two parts: classification and regression, where classification is mainly used to predict discrete class labels, such as true or false, and others, and regression is used to indicate the continuous class label. The models used were random forest classification, K-Nearest Neighbor, and logistic regression. The reasons are that KNN is very simple to implement and is non-parametric. Logistic regression is easy to interpret and has probabilistic output. Finally, the random forest has high accuracy and can handle complex datasets, preventing overfitting.

5.2.4 Mobile Application

5.2.4.1 Vital Signs Dashboard

The dashboard was designed and built using Visual Studio Code, Flutter, and Django backend with the Firebase database. The Appendix contains the code implementation. Below is a sample of the dashboard data in Figure 5.8.





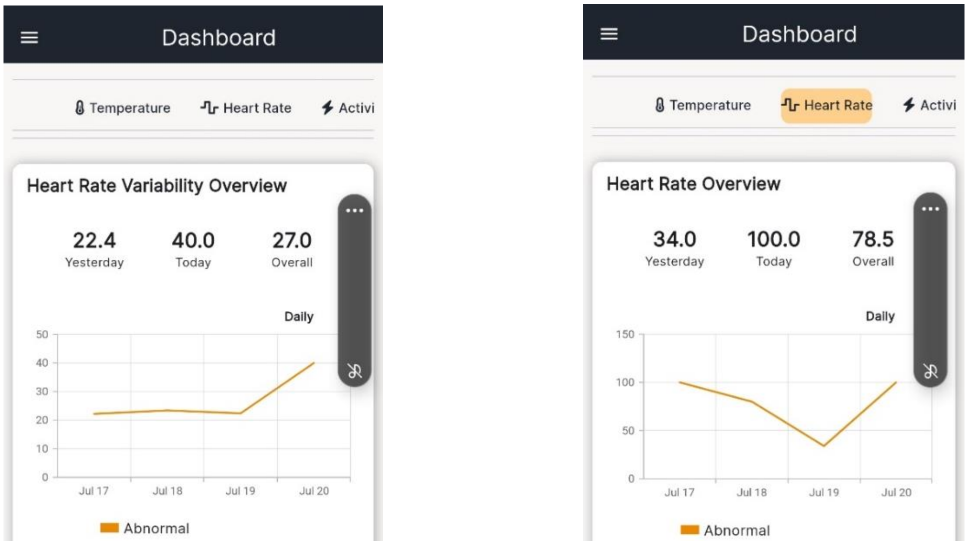


Figure 5.8: Mobile App Dashboard

5.2.4.2 Health Tips and Notification

Health tips were added to the mobile application to give pet owners health tips based on the pet's health status. It is meant to educate pet owners about what to do when they contact a veterinarian. These health tips can be referred to when there are abnormalities in the pet's health. As shown in Figure 5.9, (a) displays the status of the pet's health while trying to locate a veterinary hospital. (b) shows the health tips from health document sources.

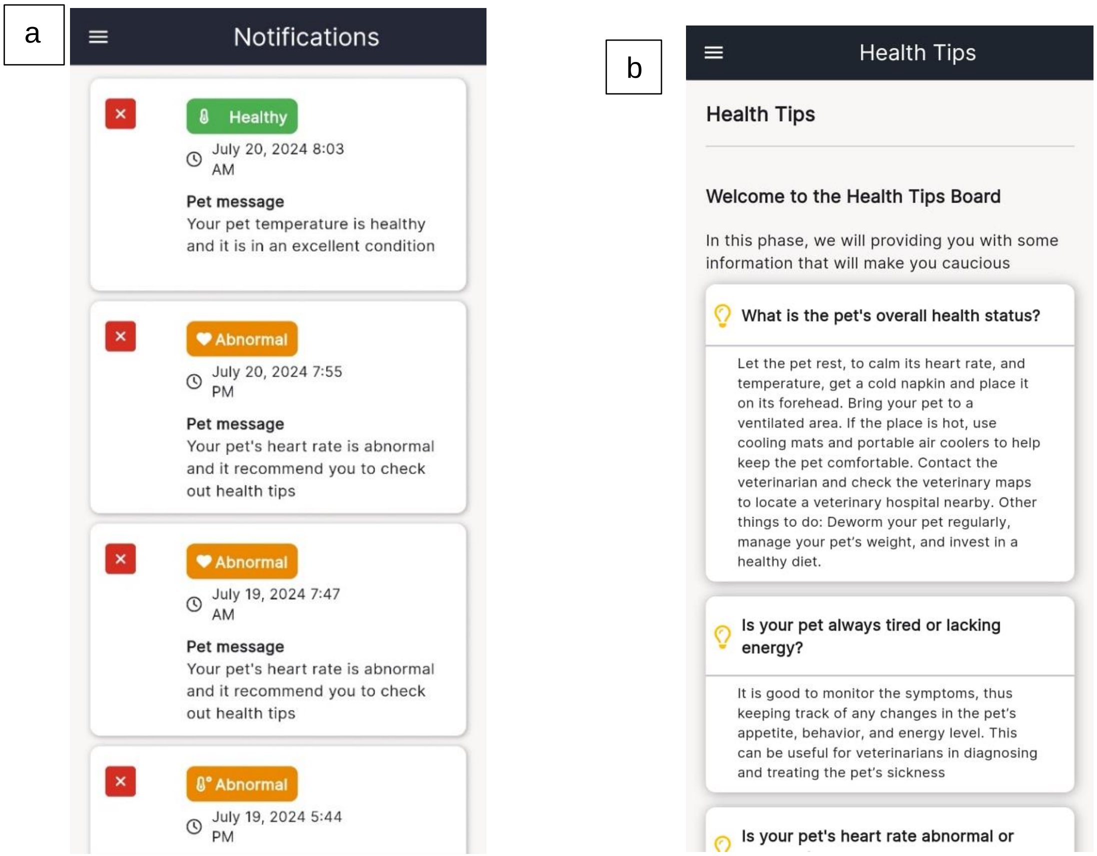


Figure 5.9: (a) Notification for pet owners (b) Health tips for pet owners

5.2.4.3 Mobile Application's maps

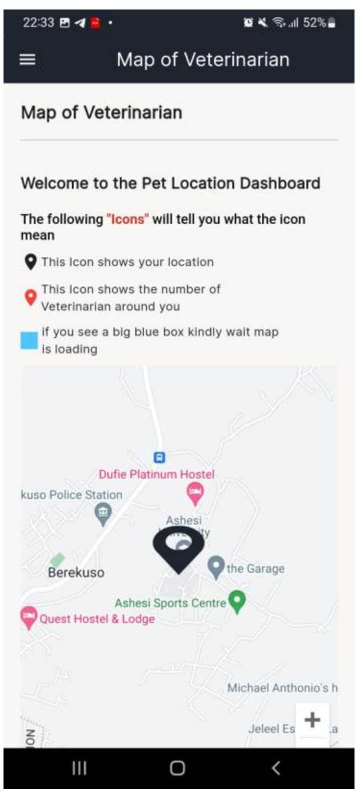
Pet owners can check for veterinary hospitals near them based on location. However, they must turn on their location so that veterinary hospitals near them can be shown on the map. The veterinary hospital map was implemented using Google API. An API key was obtained from the Google Cloud platform to authenticate the mobile application. The 'g**oogle.maps.Map'** object was used to initialize the map and specify the DOM element and map options, such as the zoom level and the center coordinates, where the coordinates were placed inside, including length, radius, and latitude. Additional objects, '**google.maps.Marker'**, were used to add shapes, markers, and routes to the map. Event listeners were then attached to map elements like clicks or drag for user interaction; the code can be seen in Figure 5.10.

```
    // Get nearby places using the current location
    _getNearbyPlaces(position.latitude, position.longitude);
  }
```

```
  Future<void> _getNearbyPlaces(double lat, double lng) async {
    PlacesSearchResponse response = await places.searchNearbyWithRadius(
      Location(lat: lat, lng: lng),
      5000,
      type: 'veterinary_care',
    );
```

Figure 5.10: Code of how veterinary hospitals were retrieved on the map.

The coordinates are taken from the database for the pet's location or pet owner's mobile or web location, but permission is asked first before it fetches those data. Finally, it displays all the veterinary hospitals surrounding the pet owners. However, for the pet location, it just showed where the pet is currently. Before one can view their pet's location, they must add a pet.



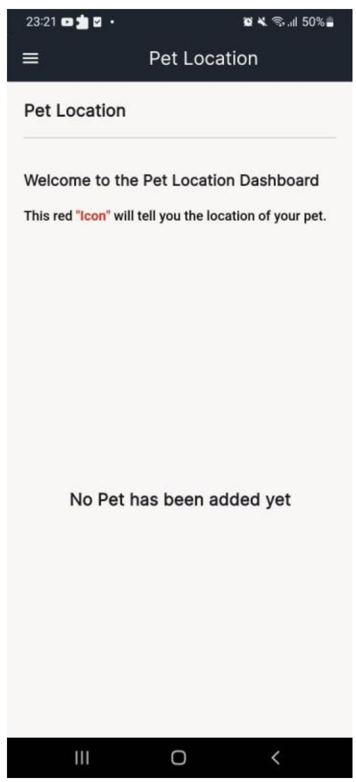


Figure 5.11: Maps of Veterinary Hospital and Pet Location

5.3 Integration of All Parts

The contextual diagram illustrates how the hardware and software implementations were integrated to create a more comprehensive depiction. This would guarantee that no component is unrelated to the others. By pressing an app, the hardware device can be turned on or off using the Kelpet app. However, before pet owners can view their pet's health information, they must have a serial number, such as a smart collar serial number. Pet owners can add up to 5 pets, but all must have different smart collar serial numbers. If they wish to use the previous smart pet collar for another pet, the previous pet's information will be cleared from the database, and after that, the new pet can now own the smart pet collar. After completing the following procedures, the project hardware could now be packaged in an appropriate package:

- PCB designing, printing, and soldering.
- 3D printing of the container.
- Coding of database.
- Design of the Kelpet App.

a

b

Figure 5.12 Fully functional smart pet collar – Kelpet

Figure 5.12 shows the whole design of the smart pet collar after testing the codes and designs. [a] is a comfortable crocheted collar strip, and [b] is a smart pet collar with belt strips.

Chapter 6: Testing, Result, and Analysis

6.1 Introduction

The system aims to provide pet owners with a method to track their pets' health and help them locate their pets and veterinary hospitals while giving them health tips and prompt warnings if their pets' health is abnormal. Therefore, the system should be reliable. The exact areas on which the testing was concentrated were as follows.

- Mobile application.
- Live Sensor readings on the dashboard
- Mobile application conditions for health status, Google Maps.

6.2 Numerical Testing and Analysis

This section's introduction of proper statistical tests confirmed the product's success. Further testing was done on the actual vital sign values of the pet versus the designed sign values recorded, thus, heart rate and temperature readings.

6.2.1 Test on the temperature sensor

One vital aspect of the system is precision production. Accuracy is relevant in the system, as it deals with the health of pets; therefore, the values recorded should be accurate and precise based on the pet's health status. The outcomes of measuring the ten input temperature values in the simulation are displayed in the table below.

Table 6.1: Results obtained from temperature sensor simulation

Percentage Error = $\left|\frac{output-Input}{input} * 100\right| = |\frac{error}{input} * 100|$

Test	Input (∗C)	Output (∗C)	Errors	%Error (%)
1	38.3	38.3	0.0	0.0
2	38.4	38.3	-0.1	0.26

3	38.9	38.8	-0.1	0.26
4	39.2	39.1	-0.1	0.26
5	39.4	39.3	-0.1	0.25
6	39.8	39.8	0.0	0.0
7	40.1	40.0	-0.1	0.25
8	40.2	40.1	-0.1	0.26
9	40.5	40.1	-0.1	0.25
10	40.8	40.6	-0.2	0.49

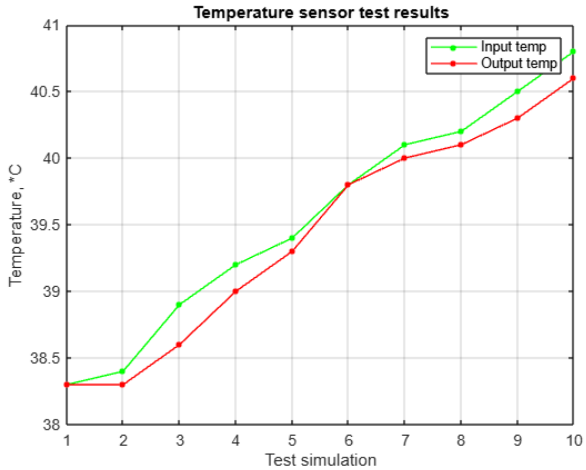


Figure 6.1: Plot of temperature against values from simulations.

<u>Analysis</u>

Figure 6.1 shows that the output temperature of the smart pet collar closely follows the input temperature with minor deviations. Most errors are negative and range from -0.2 to 0.0, showing that the output was slightly lower. Also, from Table 6.1, all the error margins were below 5%, satisfying my system requirement of accuracy, which should be at most 5%. Therefore, the readings are acceptable, showing their reliability for monitoring the pet's temperature level.

6.2.2 Test on the Activity Level Evaluation

After calibrating the MPU6050 to ensure accurate acceleration measurement and getting the baseline by placing the smart pet collar on a flat surface, it was attached to the pet. The data collection aimed to capture various activity levels, thus inactive, moderate, and active.

6.2.2.1 Data Analysis

Data was collected and then preprocessed. Consequently, the unprocessed accelerometer data was cleansed. Missing value handling, noise filtering, and data normalization were required to maintain consistency. Relevant features were extracted from the accelerometer data, including the acceleration magnitude and individual readings along the X and Z axes. The magnitude was calculated using this $magnitude = \sqrt{x^2 + y^2 + z^2}$.

Machine learning algorithms, specifically K-Nearest Neighbor Logistic regression and Random Forest classification, were used to classify the activity levels. The models were trained and tested on labeled datasets to ensure accurate predictions. GridSearchCV was used to optimize the performance of the Random Forest and to optimize hyperparameter tuning. This helped select the best parameter combination to avoid overfitting and improve accuracy.

6.2.2.2 Results

The analysis yielded the following key insights; thus, the random forest accurately classified pet activity levels. The accuracy scores for both testing and training models are shown below.

a

```
Model performance for Training set
- Accuracy: 0.9971
```

b

```
Test Accuracy: 0.9986301369863013
```

Figure 6. 2: (a) Accuracy score of the training models (b) Accuracy for testing models

Additionally, the confusion matrix for the Random Forest model showed that most misclassification occurred between **inactive** and **mini-active categories,** suggesting a need for further refinement in differentiating these activity levels. For the feature importance, it showed that the magnitude of acceleration was the most crucial predictor of activity levels, followed by the X-axis readings.

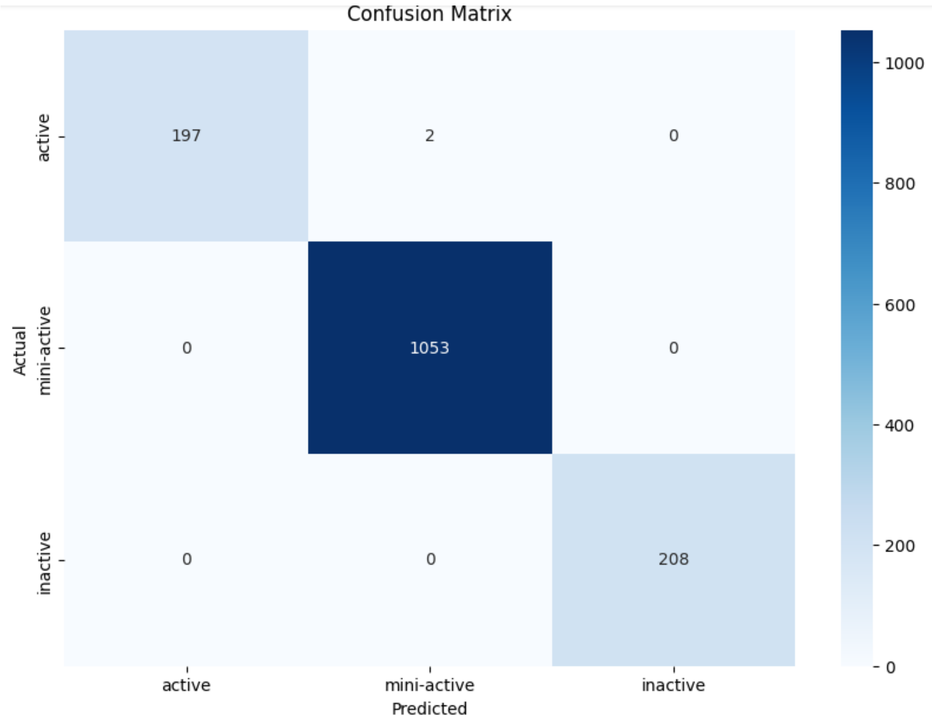


Figure 6.3: Confusion matrix for the activity levels

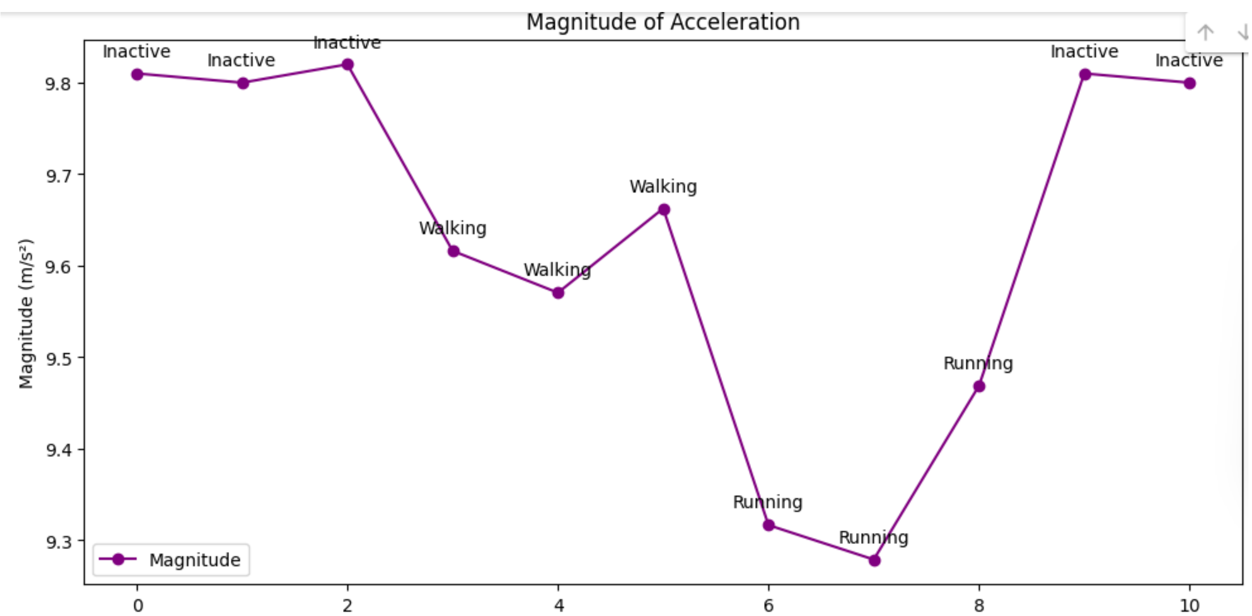


Figure 6.4: Plot of some data from the sensor reading for the activity level

6.2.2.3 Interpretation of Results

The testing and analysis of the MPU6050 showed the effectiveness of the smart pet collar in monitoring pet activities. The high accuracy of the Random Forest model showed its suitability for real-time activity classification. However, the misclassification between the active and mini-active suggests the need for new features or more sophisticated algorithms to enhance precision.

Table 6.3 Test on the Heart Rate Data

Test	HR_{input} **(BPM)**
9:00	140
9:15	142
9:30	141
9:45	138
10:00	140
10:15	143
10:30	139
10:45	140
11:00	141
11:15	138

Analysis of heart rate trend after testing

The heart rate data was recorded at 15-minute intervals over 3 hours. The heart rate values were consistent between 138 bpm and 143 bpm. Based on veterinary standards, a cat's typical resting heart rate ranges between 140 bpm and 220 bpm. The recorded heart rate data falls within the normal range for a resting or slightly active cat, indicating the cat is healthy.

a

```
- Accuracy: 0.9947
- F1 score: 0.9947
- Precision: 0.9947
- Recall: 0.9947
- ROC AUC: 0.9999
```

b

```
- Accuracy: 0.8304
- F1 score: 0.7955
- Precision: 0.8514
- Recall: 0.8304
- ROC AUC: 0.8066
```

Figure 6.5: (a) health prediction – Random forest (b) health prediction – Logistic regression

Reason for the Logistic regression accuracy: The target label has no linear correlation with the features. In such a situation, logistic regression cannot accurately predict targets, even on the training data.

6.3 GPS Overall Testing

6.3.1 GPS testing – Pet Location

GPS tracking implemented in the smart pet collar functionality was tested after setting up the hardware and software. The main objective was to accurately determine the pet's location using the ATGM336H GPS module, part of the smart collar, and to ensure the data was correctly transmitted to the pet app for real-time location tracking. The ESP32-S2 microcontroller processed the GPS longitude and latitude. It transmitted it to a remote database via Wi-Fi and MQTT and displayed retrieve on the pet Google map. Testing was conducted at Ashesi University, ensuring a clear line of sight to satellites for optimal GPS signal reception. The GPS module was initialized with the collar powered on and displayed on the mobile app.

6.3.2 GPS testing – Veterinary Hospital

Testing was conducted at Ashesi University and Madina, ensuring a clear line of sight to the satellite for optimal GPS signal reception. This was done to compare whether the GPS can show if there are veterinary hospitals close to Ashesi and whether there are veterinary hospitals in Madina.

6.3.4 GPS results after testing

The GPS consistently recorded the coordinates corresponding to Ashesi University's Latitude. $\mathbf{5.75892}^{\circ}\boldsymbol{N}$ longitude $\mathbf{-0.2238}^{\circ}\boldsymbol{N}$**.** The app accurately showed the pet's location on a map, pinpointing Ashesi University. The map interface in the app provided a clear and precise representation of the pet's location, which can be seen in the figure below. Red means veterinary hospitals available or the pet's location. At the same time, Green means my current location.

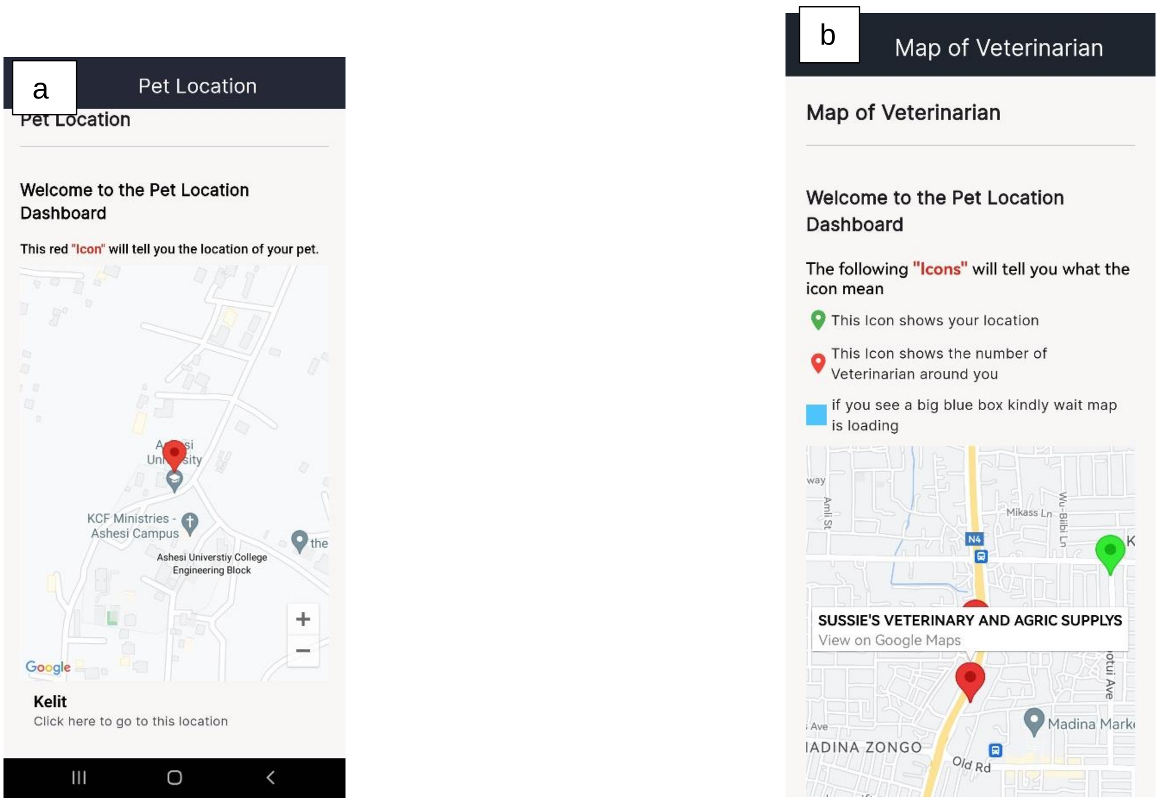


Figure 6.6: (a) Map of the pet's location. (b) Veterinary Hospitals at Madina

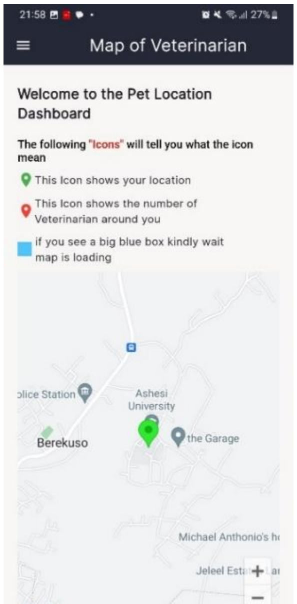


Figure 6.7: No Veterinary Hospitals close to Ashesi University

6.3.5 Analysis of the GPS Trackers

The ATGM336H GPS showed high accuracy in determining the pet's location, consistently matching the known coordinates of Ashesi University. The integration with ESP2-S2 and the use of MQTT over Wi-Fi proved effective for real-time data transmission and retrieval. The same was true for the veterinary hospitals, as it was determined that there were no veterinary hospitals close to Ashesi University. Still, the veterinary hospitals were pinpointed when it was tested at Madina. Regarding reliability, the collar recorded and transmitted data without significant interruptions. The pet app's ability to display the correct locations verified the entire system's integrity.

6.3.6 Overall Testing of Hardware and Software – Kelpet

Testing of the hardware was done at Ashesi University. This was done by placing the collar on a pet's collar and recording the values of the pet's health.

a. b.

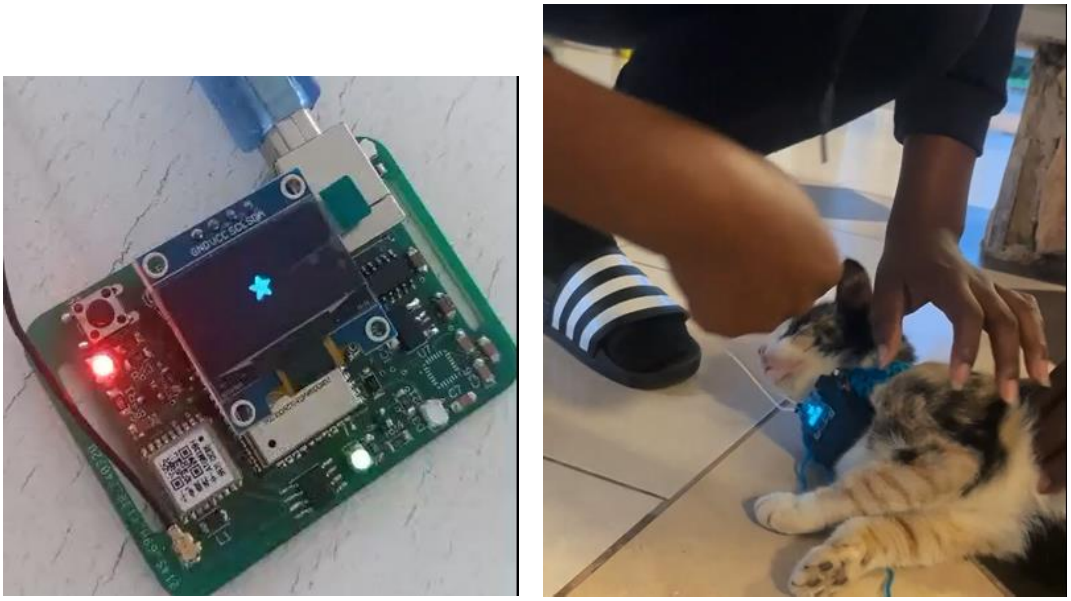


Figure 6.8 Testing the smart pet collar on a cat at Ashesi University.

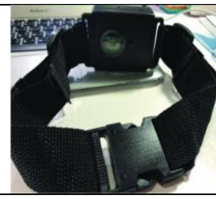
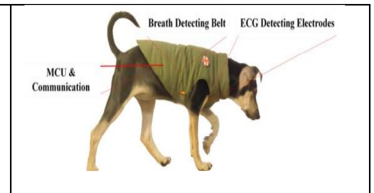
[a] was when testing of all components was working. [b] was when the overall testing of the smart pet collar and the software at Ashesi University. The mobile app could show the pet's health metrics, track the pet's locations, send notifications, and even give some health tips. However, to use the software thoroughly, check [Appendix 5] for the QR code for the Kelpet web app. One must have the smart collar serial number embedded in the hardware code. Thus, add a pet and use the full functionality of the software. For the mobile app installation, check Appendix 5 as well.

6.4 Comparison of the new system, Kelpet, with other existing systems

The new system was compared to the existing devices discussed in Chapter 2. Table 6.4 shows the performance, cost, features, and overall efficiency difference. From Table 6.4, the new system had more improvements, including in terms of cost, than the other existing devices.

Table 6.4 Comparison of the Kelpet system to other existing devices

Descriptions	**Kelpet system**	**Sick animal health monitoring**	**Smart Pet Clothing**

Sketch			
Criteria			
Cost	Hardware costs $25, and software costs $5, making $30 overall. The yearly cost is $85	The average cost of $119 plus software is $134. Yearly fee is $299.	Cost $100
Battery Performance	Lasted for 5 hours and 10 minutes.	Lasted for only 4 hours.	Last for 5 hours
Vital signs measurement	Measure heart rate, activity levels, temperature	Heart rate, temperature, and humidity.	Measure HRV, HR, and temperature.
Mobile app features	Health tips, notifications, and dashboards, and pet owners can add more than one pet.	Dashboard, notifications, and pet owners can only register for one pet at a time.	Dashboard for monitoring pet health, chat box for contacting veterinarians as well.
Secure data transmission	User and pet information are secured.	User and pet information are secured.	User and pet information are secured.
Google maps	Maps for veterinary hospitals and pet locations.	No Google Maps.	Maps track pet locations but do not locate veterinary hospitals.
Interruption with sensors' accuracy	There are low interruptions since the sensors are secured.	There are low interruptions since the sensors are secured.	It will likely affect sensors when washing or when a pet rolls on the floor.

From Table 6.4, Kelpet had more functionalities than the existing ones and accomplished almost 98 percent of the project objectives and scope. This shows the project's success since nearly all the needed implementations were done within the given timeline, thus one year.

Chapter 7: Conclusion and Future of Work

7.1 Conclusion

The project aimed to design a health monitoring and analyzing system for pets. It was established by going through different phases of design processes after doing research and literature reviews. Those different phases made it possible to build a smart pet collar that records pet vital signs and a mobile/ web application that shows the pet's health, locations, and veterinary hospitals around the pet owners. The project was meant to help African pet owners with lower-cost pet health devices; it has proven that an embedded system with the Internet of Things will help create a better system with more features. The software part worked correctly, as it was able to show the health status, the pet locations, veterinary hospitals in an area, allow pet owners to add more than two pets with the collar serial number, update the pet's information, show various health tips for difference occasions and finally log in and log out from the system.

All parts of the collar performed as expected. The sensors were able to fulfill the purpose of as it was presumed. The DS18B20 showed an accuracy of 98%.

7.2 Limitations

Even though the system performed its functionality-based work scope, its implementation was complex. These ultimately affected the overall efficiency of the product.

1. Booking veterinary appointments was not implemented because of doctors' permission.
2. The power consumption mechanism was meant to be implemented, but because real-time health data was a priority, the sleep mechanism in ESP32 still needed to be implemented.
3. A limited dataset about pet diseases affected extra features to help with disease predictions.
4. The device was limited to the health status of cats and dogs only.

5. Ensuring stable Wi-Fi connectivity in varying environmental conditions was challenging for the GPS trackers. There is a need to minimize the latency of data transmission and retrieval.
6. The Nokia battery was used instead of the LiPo battery because of its unavailability.

7.3 Future Work

Future directions for the mobile application design and the intelligent pet collar are covered in this section. After investigating measures to enhance the designs' limits, future work can be put into practice.

A solar panel and battery will charge the lithium polymer battery as backup power or a charging module when the day's electricity is unavailable.

- Getting permission from veterinary doctors and implementing a section for the doctors to register and have an appointment with an owner who has booked an appointment. Pet owners also have a site that they can use to book an appointment.
- They will use the chat box to navigate the app and contact us for further information.
- Potential improvement for the GPS Tracker: Implement power-saving techniques for the GPS module during periods of inactivity. Additionally, the user interface of the pet app should be enhanced to include additional features like geofencing and location history tracking.

References

[1]A. Zárate, A. Espinosa, C. Roberto, and N. Calvo, “Heart Monitoring Vest (MAC) for Cats and Dogs,” *IFMBE Proceedings*, pp. 365–373, Oct. 2023, doi: https://doi.org/10.1007/978-3-031-46933-6_38.

[2]“Web Application for Sick Animals Health Monitoring System | IEEE Conference Publication | IEEE Xplore," *ieeexplore.ieee.org*. https://ieeexplore.ieee.org/document/9090744 (accessed November 16, 2023).

[3]J. Chen, Y. Ren, H. Xiang, X. Jiang, X. Yang, and H. Guo, "A self-powered human-pet interaction system enabled by triboelectric nanogenerator functionalized pet-leash," *Nano Energy*, vol. 101, p. 107597, Oct. 2022, doi: https://doi.org/10.1016/j.nanoen.2022.107597.

[4]Y. Kamat and S. Nasnodkar, "Advances in Technologies and Methods for Behavior, Emotion, and Health Monitoring in Pets," *Applied Research in Artificial Intelligence and Cloud Computing*, vol. 1, no. 1, pp. 38–57, Dec. 2018, Accessed: November 26, 2023. [Online]. Available: https://researchberg.com/index.php/araic/article/view/158/151

[5]“Pet allergen levels in homes in Ghana and the United Kingdom," *full-text*, 2001. https://www.jacionline.org/article/S0091-6749(01)08811-X/fulltext (accessed 2002).

[6]“The Best Pet Cameras for Monitoring Your Dog or Cat | Preventive Vet," *www.preventivevet.com*. https://www.preventivevet.com/pets/the-best-pet-cameras (accessed January 14, 2024).

[7] ESP32 MCU Development Board Price In BD | RoboDoc. https://robodocbd.com/product/esp32-development-board

[8] . https://diyodemag.com/projects/kids_basics_automatic_lighthouse

[9] How OLEDs (organic LEDs) work - Explain that Stuff.

https://www.explainthatstuff.com/how-oleds-and-leps-work.html

[10] "Petpace Health 2.0: Next-Gen Pet Health Monitoring," Dec. 05, 2014. https://petpace.com/petpace-hits-the-mark-with-smart-collar-for-dogs/ (accessed Jul. 30, 2024).

Appendix

Appendix 1: Interview Questions

Pet health monitoring system and analysis device questions

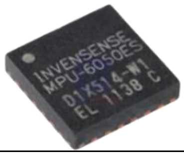
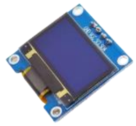
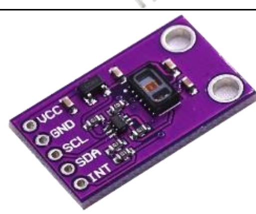
Question to some pet owners:

1. How often do you get worried about your pet's health?
2. When did you last send your pet to the veterinary hospital?
3. How often do you send your pet to the veterinarian for checkups?
4. How many veterinary hospitals are nearby or in your community?

Questions to some veterinarians

1. Do you find your work overcrowded or not?
2. How frequently do pet owners bring their pets for checkups?
3. What vital signs do you expect smart pet health devices to have?

Appendix 2: Images of the electrical components

Electrical components	Images	Name
MPU6050 chip		Accelerometer sensor
0.9-inch OLED		Display screen
AMS117 -3.3		Voltage regulator
ATGM336H		GPS module
DS18B20		Temperature sensor
MAX30105		Heart rate sensor
6 x 6 x 1 Push button four pins		Push button switch
Type C USB		USB Type-C

SMD LEDs		Surface mounted LEDs
LiPo Battey 3.7V 2000mAh 7.4Wh		Lithium Polymer Battery

Appendix 3: Electrical 3D model of the PCB design

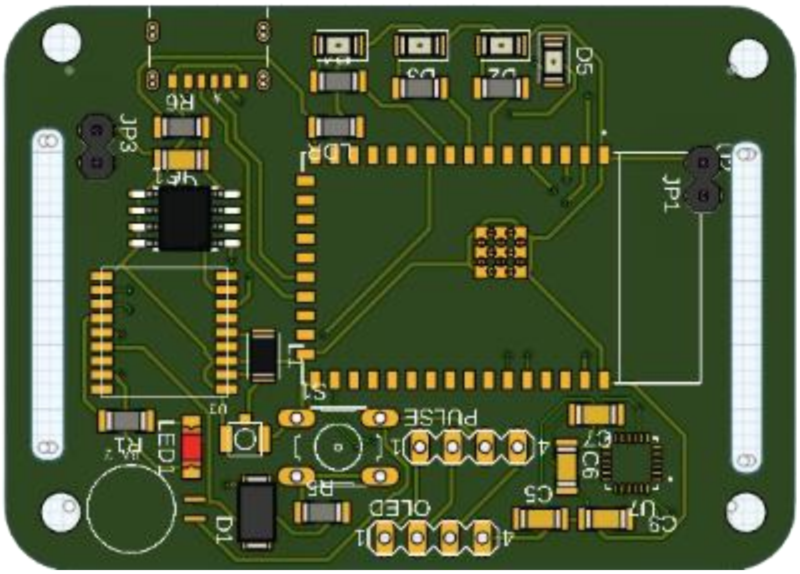


Fig: Front-view of the 3d model for PCB design

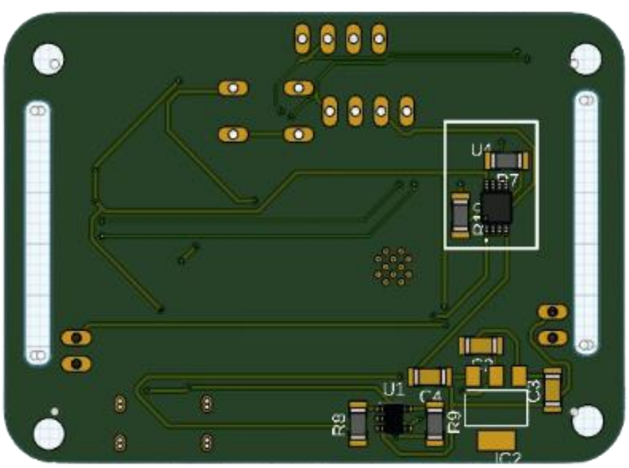
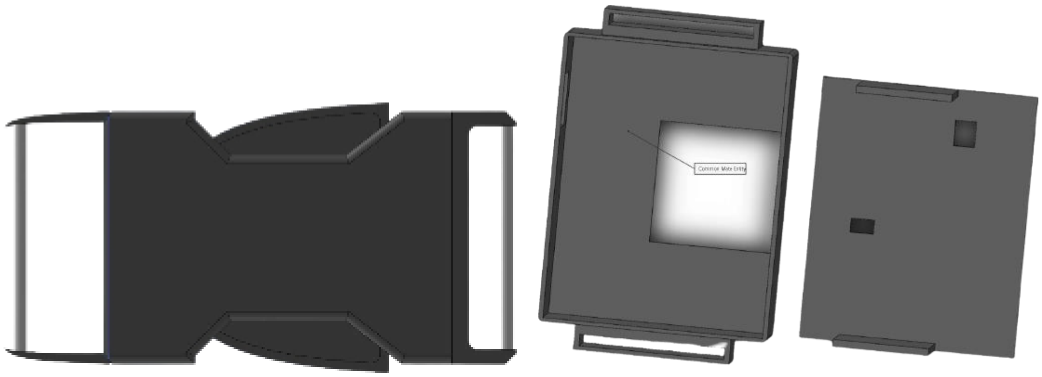


Fig: Back-view of the 3D model for the PCB design

Appendix 4: SolidWorks Design of Parts



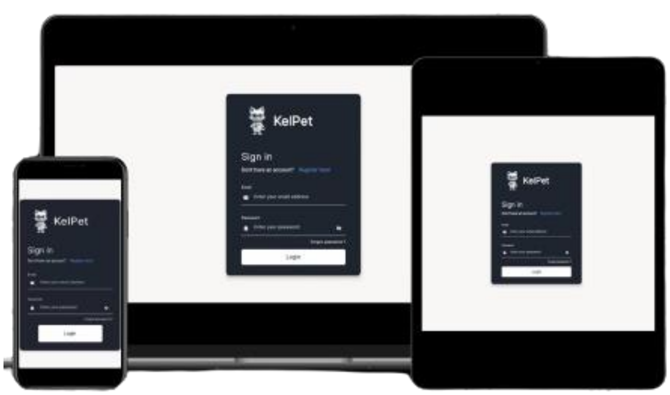
Appendix 5: Datasets used for training

[Dataset about Accelerometer readings for dogs]

Appendix 6: QR to Kelpet Web Application and Other Forms of Application

Kelpet web application QR code

Kelpet Mobile App – From my Github Account



The different forms of kelpet application usage