ECE 445 DESIGN DOCUMENT : PET HEALTH MONITOR
Team 59

Rushil Shah (rushil2)
Jeffery Haag (jrhaag2)
Tanmay Thakur (tthakur2)

1.Introduction

1.1. Problem

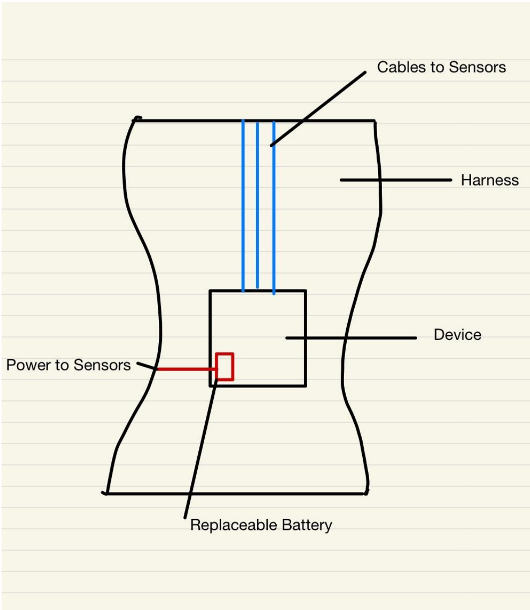
A pet owner's prime concern is the well being of their pets even though the trip to a veterinary doctor is usually very expensive. According to various pet care blogs the average cost of a routine check up can end up anywhere between $50 - $250 depending on the pet, tests required and the doctor's medical opinion. A short term hospitalization can cost anywhere between $600 - $1700. In order to prevent a serious complication, it is very important to monitor the overall health of one's pets.

In order to help alleviate a part of this cost and ensure that an owner has a holistic view of their pet's health at a low cost, we propose a wearable harness that allows pet owners to keep track of important data during play sessions or as they go about their daily routine.

1.2. Visual Aid

Below we see images of a harness which would resemble the harness our device would be fitted into. We wish to install the circuit board on the back of the harness, near where the leash is connected. The board will be covered with a soft material to cover exposed sharp edges. We expect the cables to run along the straps of the harness and they will be covered too. We see how our design would fit onto a harness below too.

Figure 1 & 2: Physical Design and Use of a Harness



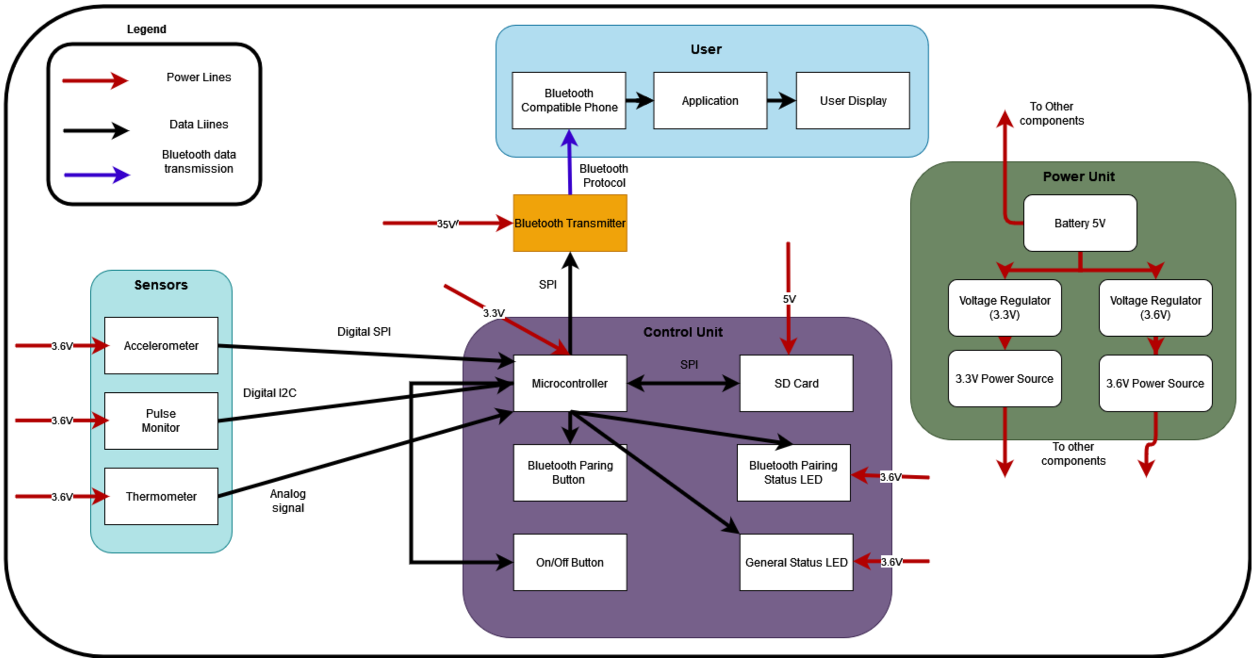
1.3. High Level Requirements:

- Sensors should deliver a stream of data to the control unit at-least once every 10 seconds.
- Memory is updated with sensor data during each session storing a maximum of 24 hours' worth of data. When the user connects, the memory device is flushed.
- Device can maintain wireless connection to the user's smartphone and is able to transfer memory contents to the smartphone.
- A user's smartphone can perform computations on incoming data and present accurate readings to the user in a human interpretable format.

2. Design

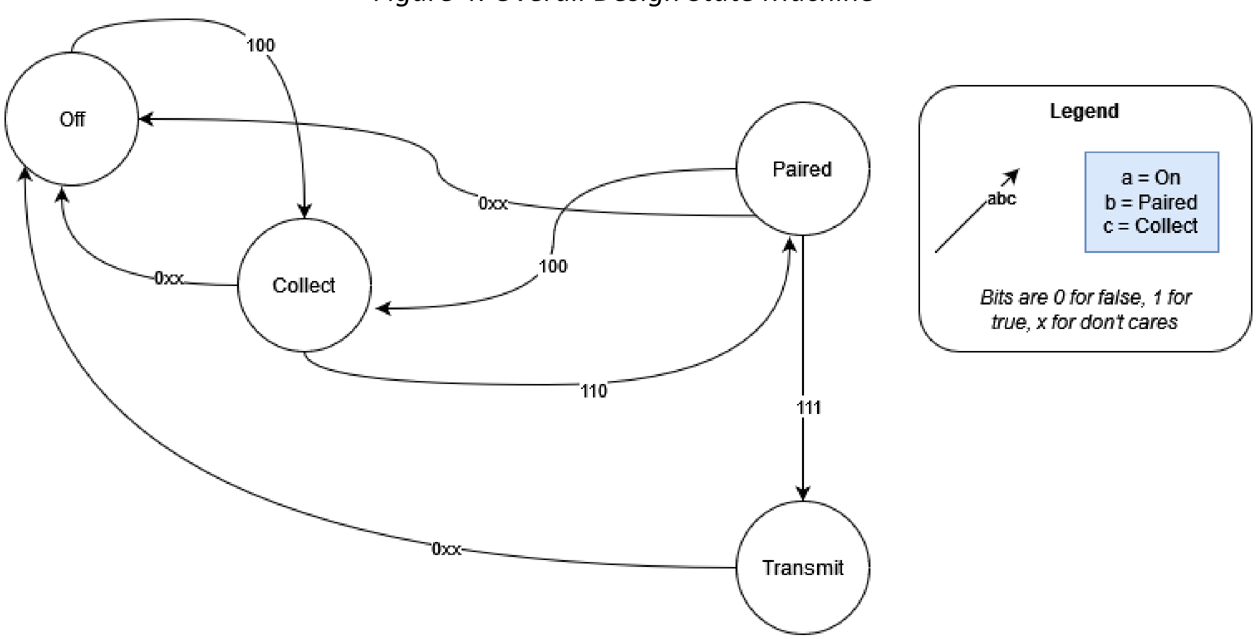
2.1. Block Diagram

Figure 3: Overall Block Diagram



Below is a state machine for the entire device's functionality.

Figure 4: Overall Design State Machine



As seen above, we start in the off state. We return to the off state whenever the On signal is 0 from any other state. The Off state always takes us to our default state - Collect. Here, bit packets from the sensors are stored into the SD-Card at the rate they are collected. When we receive the Paired signal we move to the Paired state, where we are ready to transmit data to the user-end and only wait for the transmit notification. Finally, in the transmit state, data from the SD-card on the board will be transmitted to the smartphone where our algorithm parses the data.

2.2. Subsystems

2.2.1. User Subsystem

Input: Bluetooth data incoming from transmitter
Output: Data presented on the app
The user system entails a smartphone running an app of our design. The smartphone will have bluetooth compatibility and an android app will run on top of it. By using an android smartphone we make it much easier for the user to access their animals data as opposed to some other receiving unit. Once the data has been downloaded the app will do some arithmetic and other data processing in order to give the user accurate sensor data in a human readable format so the user doesn’t need to interpret raw analog/digital data.

The temperature sensor information will be converted to a temperature in celsius as follows.

$$T = \frac{V - 0.0075}{0.100}$$

Where V is the voltage on the dataline. This is an analog signal of course, but a measurement will be recorded every clock cycle. The application will average out the T’s for an average temperature over a minute’s time. This will then be plotted on a graph, and compared with the healthy value for a cat. Calculating the time outside range is trivial.

To find the distance traveled, we will receive data for (x,y,z) acceleration for every clock cycle. We will then average these values out to get an average for each 100ms increment of time. Using this acceleration and this time, distance(x,y,z) can be calculated as

$$D_x = \frac{1}{2} a_x \cdot t^2$$

Where t is 0.1 seconds as previously mentioned. The D_y and D_z are similarly calculated with a_y and a_z respectively. Once all are found total distance is calculated,

$$D = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$$

This is of course summed over the course of the period to give a distance traveled measurement.

Pulse monitor is the simplest, values are simply averaged over a 1 minute interval and graphed. The time outside healthy range is (simply times outside range * 1 minute) / total time = Percent outside range.

Below is the Requirements and Verification table for the User Subsystem

R&V Table 1: User-End Subsystem

Requirements	Verification
Smartphone is able to connect to bluetooth unit	1.User will navigate to their smartphone's bluetooth settings and search for available devices while standing within 10 feet of the device. 2.The user will attempt to pair with the device on their phone. 3. When the smartphone returns a pair complete, or similar message it is a success.
Phone can receive data from the last 24 hours	1. After successful pairing, navigating to the smartphone app and updating the app via an on screen prompt should result in the presence of data from the devices sensors dating back 24 hours. The presence of data visible in the app dating back 24 hours is a success.
App parses data to reveal metrics such as distance covered, sedentary time, and check for healthy vital signs.	1.Once in the app, the data should be visible in the form of miles traveled, sedentary time, frequency and total time with vitals spent outside healthy range. 2. 2. Success is, The presence of this data in readable form Frequency and time outside healthy vital range accurate to ±5% real value Distance traveled and Sedentary time accurate to ±20% real value.

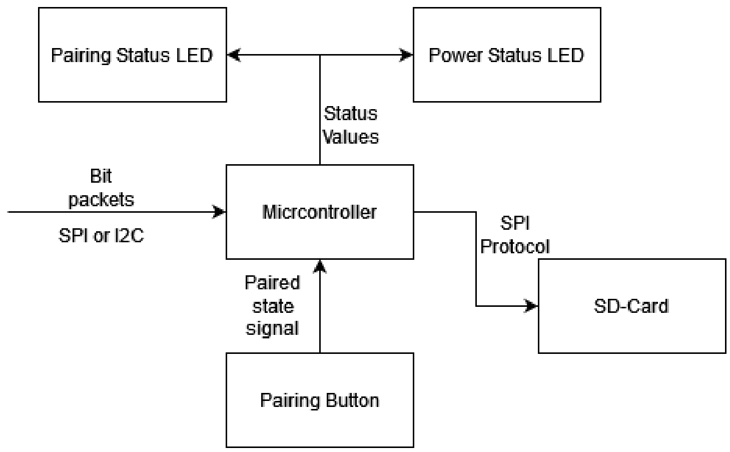
2.2.2. Control Unit Subsystem

Input: Sensor data, button input, bluetooth status

Output: Bluetooth instructions and data to transmitter, LEDs

The control unit of the system has many functions. It will receive sensory output from the multiple sensors then store it on the sd card to be given to the user down the line. It will also of course be able to read from the sd card and send information to the bluetooth module to be sent to the user system. The control unit is also responsible for allowing the user to pair their smartphone to the bluetooth unit and give feedback in the form of LED when it is trying to pair. It also has an LED to show whether the device is currently on or off.

Figure 5: Control Unit Block Diagram



Above, we have a block diagram for the control unit. Data comes in from the sensors that use SPI protocol and I2C protocol. This data is analyzed digitally by the microcontroller which communicates with the SD-Card using SPI protocol. Data will be stored in the SD-Card if this module is successfully built. The bluetooth pairing button puts the device into the pairing state, which can be seen on the LEDs.

Below we have the Requirements and Verification table for the Control Unit.

R&V Table 2: Control Unit Table

Requirements	Verification
Sensor data is received and accurately written to the SD-Card.	1. We will feed custom test data such that test data is held at a certain value to the microcontroller using a lab signal generator for digital data and a power supply for analog data. 2. After a set period we will plug the sd into a computer so we can view the raw data in .txt format.

	3.We will compare the data in the .txt to our test sensor data to ensure it is being recorded correctly
SD-Card data is read and fed to the bluetooth module	1. We will save custom data to the SD card using a computer, again such that the data is held constant. 2. We will then have the control unit output the data to the bluetooth subsystem by putting the control unit in that state. 3. We will check what data is being sent on the output line by using an oscilloscope, and verify it is the test data we saved in.
The unit is turned off and on by the button and it's state is represented by an LED light	1. We will observe the unit with a voltmeter and ensure that no wire is hot while the LED is off. 2. We will press the on button and ensure that the power LED turns on and then check with the voltmeter that the wires are now hot. 3. We will press the off button and ensure the LED turns off and the wires are no longer hot.
Bluetooth pairing state is initiated by the pairing button and is represented by an LED light.	1. The pairing LED should be off before any action is taken 2. After pressing the pairing button, the device should show as visible\pairable to any bluetooth compatible device in the area and the pair LED should turn on.

2.2.3. Power Unit Subsystem-

Input: 5V Battery Power

Output: 1. 5V ± 5% Power

2. 3.3V ± 5% Power
3. 3.6V ± 5% Power

To power our design, we would be using a 2600mAh 5V Li-ion battery as our main source. We would use this power source in parallel with 3.3V and 3.6V power sources generated by a Voltage Regulator. For 3.3V we would use a COM-00526 ROHS regulator that outputs a fixed 3.3V with internal current limiting. Additionally for 3.6V we would use a LDO voltage regulator.

Below is the Requirements and Verification Table for the Power Unit

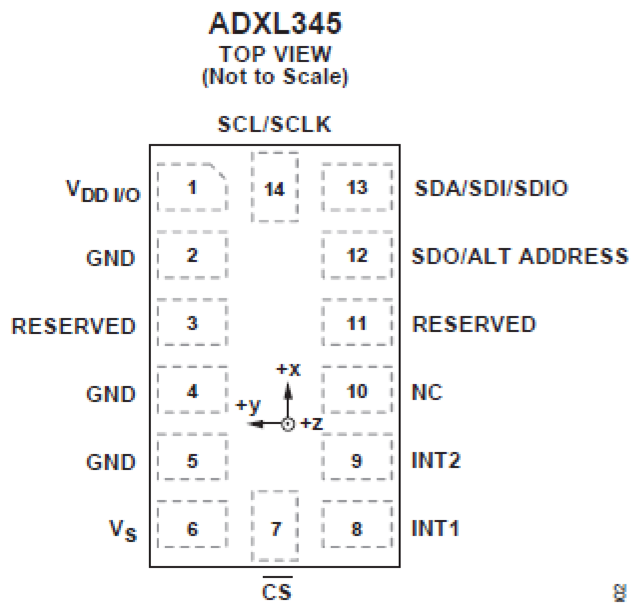
R&V Table 3: Power Unit Subsystem

Requirements	Verification
1. Battery lasts for 6 hours	1. We will measure the Potential applied and the supplied current using a multimeter and an oscilloscope while verifying individual subsystems. 2. With the recorded current and voltage values, we can calculate the power consumed by each of the subsystems. 3. Since we know the energy stored in the battery, we make sure that the battery can provide the required power for at least 6 hours.

Sensors Subsystem-
Input: Physical data collected from animal
Output: Digital signals for pulse and accelerometer and analog data for the temperature sensor to the control unit.

All our sensors receive power from 3.6V since their operating ranges fall between 2.0V at the lowest to 5.5V at the highest. We use them with 5Ω Resistors since the output current from the regulators is 0.8A, and the input voltage is 3.6V.

Figure 6: Accelerometer IC Design



Here we see an image of the Accelerometer we have chosen. Below are the assigned pins for the same.

Table 1: Schematic for Accelerometer ADXL345

Pin Number	Connection
1	3.6V Direct Interface Supply Voltage
2	GND
3	N/A
4	GND
5	GND
6	3.6V Supply Voltage from Power Unit
7	High before changing clock phase
8	N/A
9	N/A
10	No Connection
11	GND
12	SPI Output to Control Unit

13	N/A
14	Device Clock on Control Unit

Now we discuss our pulse sensor's circuit schematic.

Figure 7: MAX30102 Pulse Sensor

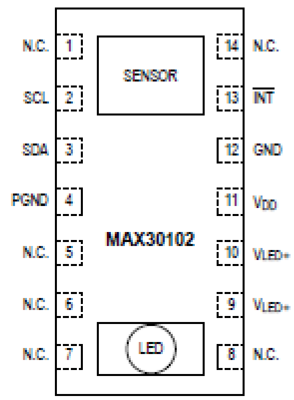


Table 2: Schematic for MAX30102 Pulse Sensor

Pin Number	Connection
1, 5, 6, 7, 8, 14	No Connection
2	Device Clock in Control Unit
3	I2C Data to Control
4	GND
9	3.3V From Power Unit
10	3.3V From Power Unit
11	3.6V From Power Unit
12	GND
13	N/A

Finally, we discuss our temperature sensor. This is the only sensor that gives us analog data, so we would need to use an ADC with it.

Figure 8: Temperature Sensor

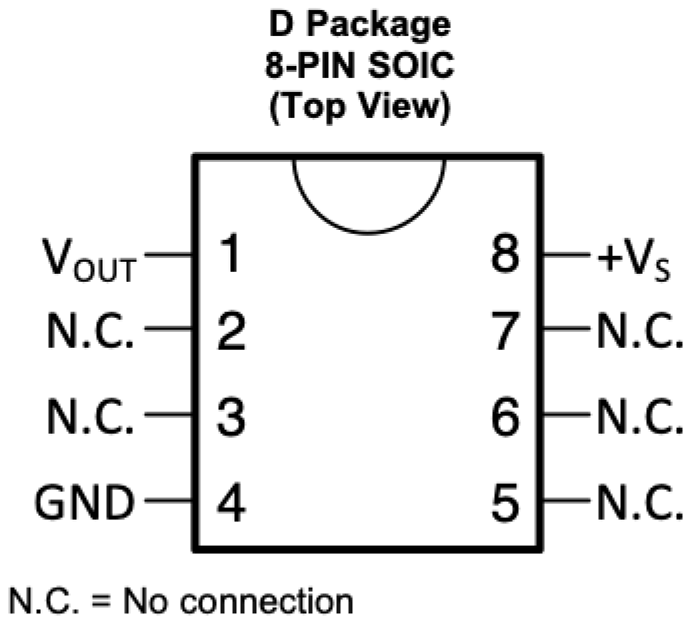


Table 3: Schematic for Temperature Sensor

Pin Number	Connection
1	3.3V From Power Unit
8	I2C data to the control unit.
4	GND
7, 6, 5, 2, 3	No Connection

Below is the Requirements and Verification Table for the Sensor Subsystem

R&V Table 4: Sensor Subsystem

Requirements	Verification
Sensors send data at at least once per second	1. We will check each sensor output with an oscilloscope and ensure it is sending information at once per second.
Sensors are active and transmitting data to the control unit, at least 95% of the time the unit is on.	1. We will allow sensors to collect data for 1 hour. Assuming the control unit has passed its earlier tests, we will then check the SD on a computer.

	2. By seeing how many data values there are on the SD, and knowing the clock speed and thus, how many values there should be, we can calculate what percent of the time the sensors were sending data. If it is greater than 95%, it passes.
The pulse sensor data and the accelerometer is accurate to ±10% of actual values and the temperature sensor is accurate to ±5% of actual values.	1. To test temperature, we will use an outside infrared thermometer and compare it to the value of the sensor recorded on the sd card at temperatures 36,36.5,37,37.5,38,38.5, and 39 celsius using the infrared as a baseline. If the average difference at each temperature is less than ±5%, it passes. 2. To test the accelerometer, we will attach a phone with an accelerometer to the cat via the harness. We will allow it to move for 10 minutes and then get the graph for x,y,z acceleration from the sensor as well as the phone. For each x,y,z we will find x = (Y_{phone} - $Y_{Sensor)}$. We will then do error = x / Y_{phone} . If x <= 0.05 it passes. 3. To test the pulse sensor, after consulting a vet, we will use a person's hand to measure the cat’s heart rate over a 5 minutes while recording pulse every 30 seconds. At the same time the sensor will be attached. We will then compare the manually gotten data to the sensor data. We will average the pulse sensor data to its value every 30 seconds. We will then calculate the differences and if it is under ±5% from the manual data it passes.

2.2.4. Bluetooth Subsystem

Input: Bit packets from SPI and I2C communication stored in SD-Card

Output: Bit packets sent to User-Subsystem

We will be using the Adafruit 2746 bluetooth transmitter module for this purpose. We propose to collect the bit packet data in the SD-Card and transmit data to the user module on request.

Below is the Requirements and Verification Table for the Bluetooth Unit.

R&V Table 5: Bluetooth Subsystem

Requirements	Verification
The unit can be paired with a smartphone and maintain a connection for at least 1 minutes.	1.We will press the pair button in the control unit, and attempt to pair it to a smartphone within 10 feet. 2. After following the smartphone prompts and the phone reports connected, we will start a five minute timer, we will monitor to make sure the smartphone reports that it is connected until the timer goes off.
Module accurately transmits data fed to it by the control system.	1.Assuming the control unit has passed it's own tests, we will load the SD card with known test data, then order the bluetooth unit to transfer data to a bluetooth compatible computer. Using Microsoft Bluetooth Serial Terminal, we will compare this received data with the preset SD card test data using a .txt /diff program.

2.3. Tolerance Analysis

Since our device cannot be plugged into power all the time, we need a battery in our design to power it. To ensure a hassle-free experience to the user, we need to ensure a reasonable battery life for the product before the battery can be replaced or charged. Since our design included several devices including sensors, radio and storage devices which need to be powered separately, we assumed our battery life to be significantly low which could lead to a reduced quality of performance.

We use the ESP32 microcontroller chip to communicate with the sensors. The chip is rated to draw 0.5 A current from a 3.3 V power supply and is the most significant source of power consumption. Another device which is rated to consume significant power is the SD card chip which uses SPI protocol to read from and write to an SD card. It is rated to draw a maximum of 0.1 A while operating on a 3.3 V source. The sensors to detect heart rate, temperature and the accelerometer consume insignificant power when compared to the ESP 32 chip and the SD Card reader. The accelerometer is rated to draw 0.023 mA at 2.5 V whereas the temperature sensor is rated to draw a current of 0.025mA at 3.3V. The PPG sensor draws a current of 0.6 mA at a 1.8V potential difference. The Bluetooth module consumes similar power as the sensors as it is rated to draw a current of just 0.0026mA at a potential difference of 3.3V.

To power our devices, we plan to use a 5 V battery rated at 2600mAh, so the total energy stored in the battery can be calculated as follows.

Energy stored in the Battery = 5 V * 2600 mAh
= 13 Watt-h

$$P_{ESP32} = 3.3V \cdot 0.5A = 1.6W$$
$$P_{Heart-rate} = 1.8 \cdot 0.6A = 1.08W$$
$$P_{Bluetooth} = 3.3V \cdot 0.0026A = 0.00858W$$
$$P_{accelerometer} = 2.5V \cdot 0.002A3 = 0.00858W$$
$$P_{temperature} = 3.3V \cdot 0.025A = 0.0826W$$
$$P_{SdCard} = 3.3V \cdot 0.1A = 0.33W$$
$$P_{total} = 3.109W$$

$$\text{Device operation time} = \text{Energy Stored in Battery} / \text{Total Power Consumed}$$
$$\text{Device operation time} = \frac{13W}{1.9812W}$$
$$\text{Device operation time} = 6.56 \text{ hours}$$

So we see that at rated power consumption for all our devices, our whole system should function for at least 6 hours as we mentioned in our requirement section of the RV table.

3. Cost and Schedule

3.1. Cost Analysis

We attempt to estimate the cost of our project by listing the main parts that we would be using and attempting to quantify our work hours. Below is a list of all the parts we expect to be using for our device.

Table 4: Cost Estimate (Parts)

Manufacturer	Model	Qty	Cost($)	Description
Adafruit	165	1	2.75	Body temperature sensor 1
Texas Instruments	LM35DZ PN JUNCTION SENSOR	1	1.66	Body temperature sensor 2
Adafruit	1093	1	25.00	Pulse sensor
Maxim Integrated	MAX30102EFD+T	1	5.06	Pulse sensor
Adafruit	1231	1	17.50	Accelerometer
Adafruit	4097	1	6.00	Accelerometer
Adafruit	387	1	0.75	LED's
Adafruit	1119	10	2.50	Buttons
Adafruit	254	1	7.50	SD Card To Spi board
Adafruit	1294	1	9.95	SD card 4 gb w/ adapter
Adafruit	2011	1	12.50	Battery 3.7V
Adafruit	1959	1	14.95	Battery 5v
Adafruit	4077	1	9.95	Bluetooth Module
Adafruit	2746	1	19.95	Bluetooth Module 2

We mention multiple sensors since there are tradeoffs to choosing each one, and the best possible fit must be found. Overall, the sum of all these prices of parts give us a worst-possible part cost, which would add up to be **$135.96 for parts.**

Table 5: Labor Cost

Labor Type	Cost/hour($)	Total Hours	Cost($)
Soldering	**40**	**40**	**1600**
Programming	**60**	**60**	**3600**
PCB Design	**50**	**30**	**1500**
Testing	**50**	**30**	**1500**
Techinical Writng	**40**	**20**	**800**
Parts Ordering	**30**	**15**	**450**
Parts			**135**
Total	**270**	**195**	**9585**

Total cost of **$9,585** and **195 hours labor.**

3.2. Schedule

2/22/2022	Today
2/28/2022	First Iteration PCB Designed
3/8/2022	Ready to order PCB Board Designed
3/9/2022	All parts including pcb ordered
3/23/2022	All sensors are operational and tested for accuracy
3/31/2022	Data can be recorded and Read from memory

3/31/2022	Smartphone App is finished
4/7/2022	Wireless module works so that Smartphone gets data
4/14/2022	Everything works and Make demo
4/18/2022	Finished Demo, Presentation
5/2/2022	Final Paper Done

4. Discussion of Ethics and Safety

Since our project is a product for animals, we must ensure their safety in the usage of the device. We will make efforts to ensure that the harness is comfortable and wearable, while taking into account that even during the testing phases there can be no harm to animals. Additionally, cats are estimated to be able to carry 2-5 kilograms (Tuxedo Cat) without issue, so 2 kilograms serves as a safe upper bound on the total weight of our harness.The Animal Welfare Act bans the use of electronic devices on animals which can stun or potentially harm an animal in any way. We need to be careful about insulating our circuits completely to eliminate accidental shocks. We also need to comply with the various articles mentioned in the 'Humane Care for Animals Act' in deciding the design of the harness and placement of the sensors. We also work in compliance with the ADA's 2002 Ethics Code through the entirety of this project. Furthermore, The FDA only requires post marketing approval for electronic products intended for animals. Additionally, for our own safety during the designing process, we would take precautions while soldering. Other aspects of our project involve little to no risk to us.

5. Citations

1. Medicine, Center for Veterinary. "How FDA Regulates Animal Devices." *U.S. Food and Drug Administration*, FDA, https://www.fda.gov/animal-veterinary/animal-health-literacy/how-fda-regulates-animal-devices.
2. Plotts, Edwin. "How Much Does a Vet Visit Cost? Here's Everything You Need to Know." *Pawlicy Advisor*, Pawlicy Advisor, 8 June 2020, https://www.pawlicy.com/blog/vet-visit-cost/.
3. PostedinHealth, et al. "How Strong Are Cats?" *Tuxedo Cat*, 25 Nov. 2021, https://www.tuxedo-cat.co.uk/how-strong-are-cats/.
4. Costa, Daniel dos S., Silvia H. N. Turco, Rodrigo P. Ramos, Flaviane M. F. M. Silva, and Murilo S. Freire. "Electronic Monitoring System for Measuring Heart Rate and Skin

Temperature in Small Ruminants." Engenharia Agrícola. Associação Brasileira de Engenharia Agrícola. Accessed February 22, 2022. https://www.scielo.br/j/eagri/a/VSYJ448gk9DqpcxyntD6MfK/?lang=en.