March 21st, 2021
Dr. Craig Scratchley
8888 University Drive
Simon Fraser University, School of Engineering Science
Burnaby, British Columbia, V5A 1S6

RE: ENSC 405W/440 Design Specification — Tenshi Baby Crib

Dear, Dr. Scratchley,

Enclosed with this letter is the Design Specification for the Tenshi Baby Crib. Our company has prepared this document to outline the design implementation so as to meet the Requirements Specification. In the document contains detailed descriptions of the product's subsystems and design choice justifications determined by the company's employees. As such, the design specification serves as a means to guide developers to implement proof of concept and prototype phase requirements.

The Tenshi Baby Crib is a high-tech crib system that monitors a baby's environment and sleeping position. By tracking temperature, noise, and the baby's movements, the system quickly notifies parents when their baby may be vulnerable so they can take action as soon as possible. To supplement notifications, our product logs and visualizes the data gathered so as to analyze environmental and sleeping patterns. With our product, parents and caretakers can ensure that their baby develops in a healthy environment.

From all of us at Tenshi, we would like to extend a thanks to you, Dr. Scratchley, and the instructional team for your continued guidance. For any questions, you may contact us by email at tenshi-company@sfu.ca.

All the best,

Chief Executive Officer, Alvin David
Tenshi Company
tenshi-company@sfu.ca

Enclosed: Tenshi Baby Crib - Design Specification

tenshi
baby crib

BABY CRIB SYSTEM DESIGN SPECIFICATION

Project Team

Dexter Bigueta
Izyl Canonicato
Alvin David
Matthew Thomas
Denyse Tran

Correspondence

Company 23
tenshi-company@sfu.ca

Submitted to

Dr. Craig Scratchley
Dr. Shervin Jannesar
Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University

Issue Date

March 26, 2021
Version 1.0.0

Abstract

At the earliest stages of life, infants are vulnerable to the environment around them. Parents and caretakers alike must constantly supervise their child's environment and sleeping position to ensure healthy growth and development. As such, taking care of a child is an enormous undertaking that requires monitoring a multitude of factors. To aid parents and caretakers in their commitment to raise their children, Tenshi Company has designed the Tenshi Baby Crib, a high-tech solution to monitor and notify the moment babies become vulnerable. In particular, the crib monitors the temperature of the room, listens for when the baby wakes up crying, and identifies when they are on their stomach or back whilst in the crib. Tenshi Company accomplished this by embedding the sensors and circuitry into the crib itself, and designing a mobile application that pairs with the crib. Once the crib has determined that the baby is in a vulnerable state, the system notifies the parents or caretakers so that they may intervene and ensure the baby is cared for. In the end, the Tenshi Baby Crib provides parents and caretakers a means to ensure healthy growth and development for their baby by efficiently monitoring their environment and sleeping conditions.

Table of Contents

Version History

Table I: Version History

Version	Author	Revision Date	Approved By	Approval Date	Reason
1.0	Tenshi Co.	03/26/21	Alvin D. CEO	03/26/21	Initial Design draft

List of Figures

List of Tables

List of Equations

Glossary

Table II: Glossary

Term	Definition
ADC	Analog to Digital Conversion
Sound Pressure Level	Pressure level of sound, measured in dB
GPIO	General Purpose Input Output
Cron job	A time based job scheduled task

1. Introduction

1.1 Background

Health Canada outlines recommendations and standards for an infant's daily active and sleeping environment in order to lessen the occurrence of vulnerable situations that would cause Sudden Infant Death Syndrome (SIDS) [1]. In particular, parents and caretakers must be proactive to follow the recommendations to mitigate the danger of overheating, prolonged distress (i.e. crying, diaper change, hunger), and suffocation. Actions taken include ideal temperature environments, responding to the baby's cries, and ensuring a safe sleeping position.

We specifically define three vulnerable situations to be unideal room temperatures, the time a baby is distressed, and sleeping whilst on their stomach. To ensure the baby is not uncomfortable or overheating, the room temperature must be between 20°C and 23°C [2]. Moreover, to quickly attend to the babies' needs, they must be attended to when they start crying. Lastly, to prevent suffocation, babies must sleep on their back. These vulnerable situations take a great deal of effort to supervise and as such the Tenshi Baby Crib aids in monitoring for potential vulnerabilities.

1.2. Technological Solution

The Tenshi Baby Crib enables parents to accurately and efficiently monitor their baby's environment and sleeping position, which is activated by placing them within the crib. To achieve this, housed within the crib are specifically designed systems that detect and measure vulnerabilities in regards to temperature, noise, and position. Especially in regards to position, a considerable amount of time must be put into innovation and design as the implementation is expected to be a challenge. When paired with a mobile phone, parents are quickly notified when their baby is at risk or crying so that they may respond with preventative measures. In addition, parents and caretakers may view visual data of their baby's environment and receive indicators and suggestions to ensure healthy growth for their children. With this, the crib offers parents and caretakers alike the peace of mind by constantly monitoring the baby, providing extra assistance with ease of access.

1.3. Scope

This document outlines the design implementation of the Tenshi Baby Crib at the proof of concept phase, and a number of prototype phase specifications. It will introduce each subsystem and its internal components. Design and structure of all subsystems will detail its adherence to the Requirements Specification. Within those subsystems, each design choice and component will be justified so as to ensure complete operation of the crib's features in an accurate, efficient, and reliable manner. Furthermore, this document also includes how the user interface will be constructed to maximize usability. Lastly, a test plan will be provided which will be used to drive development forward and sign off on the final product.

2. System Overview

The Tenshi Baby Crib consists of three subsystems: the Sensor System, the Data Analysis System, and the Notification System. These subsystems serve to measure, process, and visualize the baby's environment and sleeping conditions. When activated, the electronic sensors that make up the Sensor System measure temperature, noise, and the baby's position. These measurements are fed into the Data Analysis system to be processed and stored. If the baby is identified to be vulnerable, the parents are notified through the Notification System in the form of push notifications, graphs, indicators, and preset suggestions with a mobile application.

For the proof of concept phase of the project, the basic functionality of the crib will be implemented. Namely, the Sensor System and Data Analysis System's capability to measure inputs from the environment and detect vulnerabilities. In the prototype phase, the algorithms will be optimized, the mobile application will be fully developed and the circuitry will be physically placed in the crib. The three subsystems and their components are outlined in Figure 2.1 below.

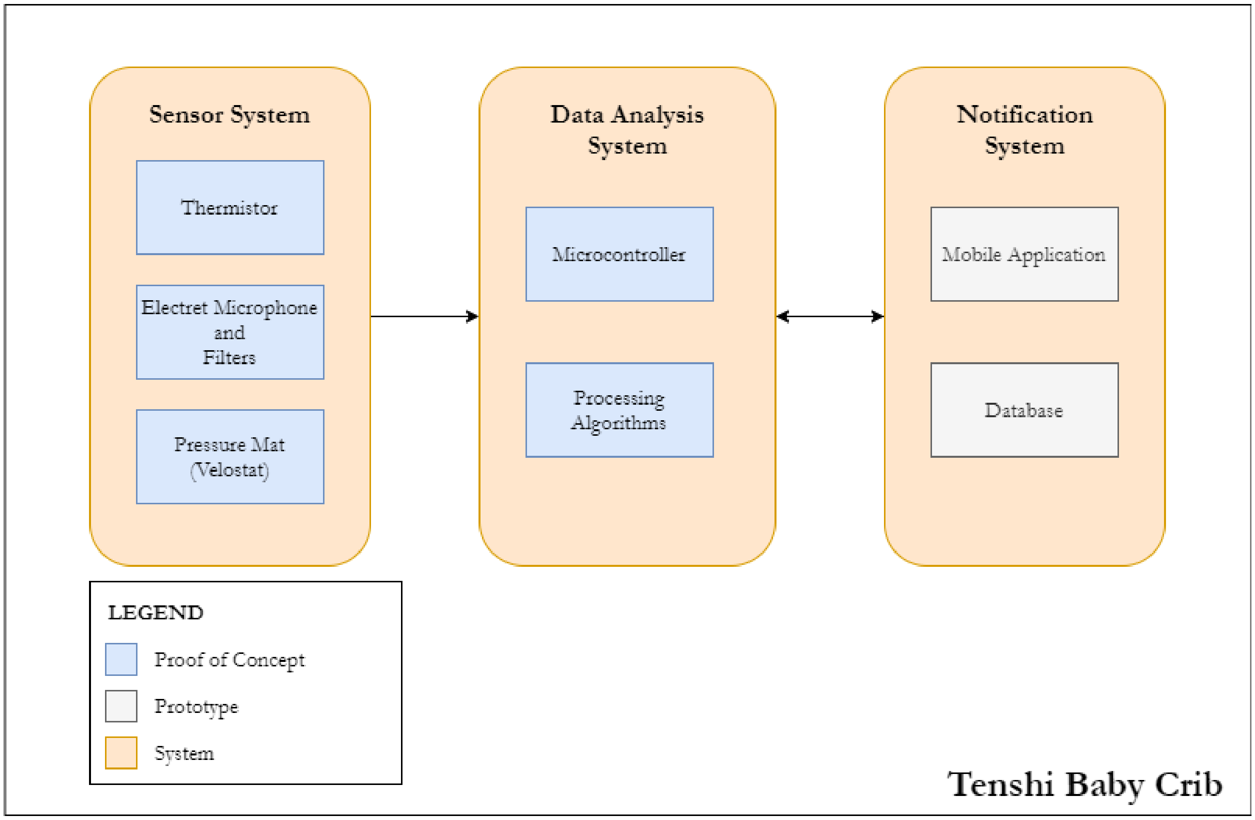

Figure 2.1: Tenshi Baby Crib Components

To further detail the plans of the prototype, the Sensor System and Data Analysis System will be contained within the crib. These two systems incorporate the electronic sensors, conductive materials, a weight sensitive material, and a microcontroller that integrates all of them together. The majority of the circuitry will be placed in the underside of the crib. A thermistor is simply placed alongside the other circuitry under the crib. Connected to the microcontroller through wiring is a sound meter which is routed through the crib's railing, so as to receive direct sound from within the crib. In addition to the sound meter, detachable wiring also connects the pressure mat in order to analyze movement. Taking input measurements from the environment and processing them, the two aforementioned subsystems communicate via internet connection to the Notification System. This system resides in the user's smartphone through a mobile application. This allows the user to always be informed about their baby's status in a timely manner. Pictured in Figure 2.2 is the intended placement of the Sensor System and Data Analysis System circuitry within the crib.

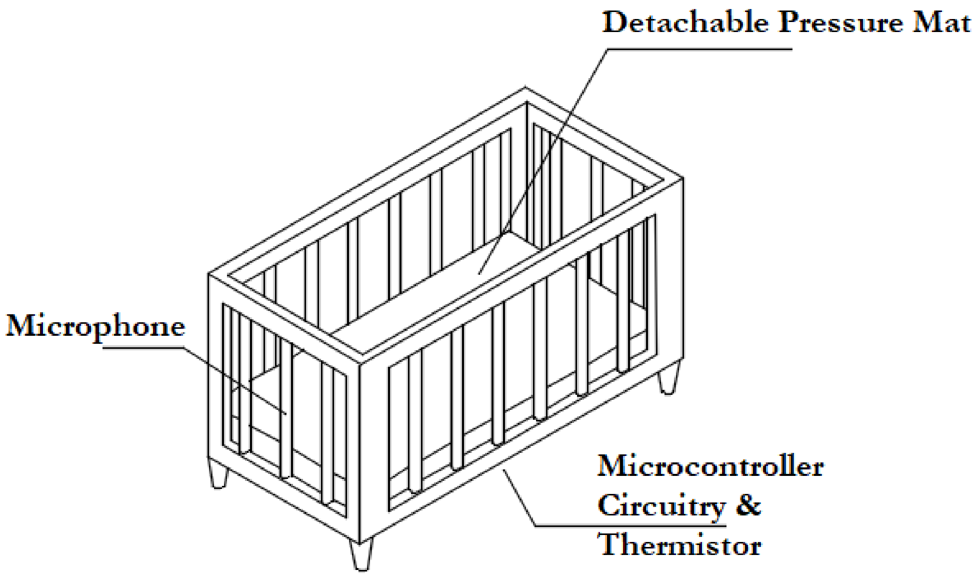

Figure 2.2: Crib System Component Placements

3. Sensor System

3.1. Temperature Monitoring

Temperature monitoring data will be gathered through a 10k NTC thermistor, with a B constant of 3950 ± 1% [3]. This thermistor has the capability to accurately report temperature to less than ±1°C as well as offer fast response times. The requirement codes are met with the chosen NTC thermistor as displayed in Table 3.1 below.

Table 3.1: Temperature Circuit Specification

	Requirement Code	Design Specification
D.3.1.1	R3.2.1-POC-1	Thermistor operating ranges from -20°C to +105°C. [3]
D.3.1.2	R3.2.1-POC-1	Thermistor is accurate to ±1°C. [3]
D.3.1.3	R3.3.1.1-PT-2	Thermistor has a fast response time and is highly sensitive. [3]

Given the temperature to resistance mapping that thermistors provide, calculations are necessary to describe the relationship. In general, NTC thermistors resistance have an inverse relationship with temperature. A voltage divider is needed as the microcontroller's ADC cannot read the voltage drop induced by the thermistor directly. The circuit is displayed in Figure 3.1.

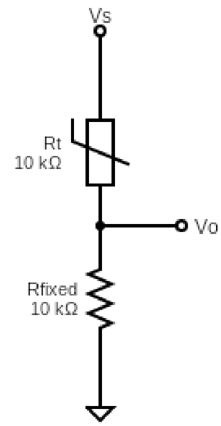

Figure 3.1: Thermistor Circuit

The voltage divider circuit will be connected to the ADC on the microcontroller through the output voltage point, V_0, as seen in Figure 3.1. The voltage divider equation as seen in (1) can be rearranged to solve for R_t, displayed in (2), which is the thermistor's resistance at ambient temperature, while R_0 is a fixed 10kΩ resistor. Additionally, V_0 is the output voltage while V_S is the source voltage.

$$V_0 = Vs\frac{R_t}{R_t + R_0} \quad (1)$$

$$R_t = R_0(\frac{V_S}{V_0} - 1)$$

(2)

Since the output voltage will be connected to the ADC on the microcontroller, the ADC_{Max} over ADC_{Val} ratio will equate to the V_S over V_0 ratio. ADC_{Max} is the 12-bit resolution value of the ESP32 which is 4095, and the ADC_{Val} is the returned ADC value through the microcontroller. As a result, the voltage ratio can be replaced with the ADC ratio shown in (3) [4].

$$R_t = R_0(\frac{ADC_{Max}}{ADC_{Val}} - 1)$$

(3)

Since thermistors are non-linear, a temperature conversion calculation is necessary to ensure accurate readings. Specifically, the Steinhart-Hart equation can be used for a close approximation of the thermistors response curve according to its operational range as followed in (4) [5].

$$\frac{1}{T} = A + Bln(R) + C(ln(R))^3$$

(4)

Since the chosen thermistor does not have coefficients A, B, and C in the Steinhart-Hart equation, which is occasionally provided by manufacturers, the equation can be simplified to a relation of temperature and resistance as seen below in (5) [6].

$$\frac{1}{T} = \frac{1}{T_0} + \frac{1}{B}ln(\frac{R}{R_0})$$

(5)

In the above formula, the variable T is the ambient temperature in Kelvin, T_0 would be room temperature, 298.15K, B is the B constant which is 3950, R_0 is thermistor resistance at room temperature and R is the resistance at room ambient temperature, which was solved above in (3). With all these values, the temperature in Kelvin can be converted to Celcius then these data points will be used in data analysis to compose an effective temperature monitoring algorithm.

3.2. Sound Detection

Sound data of the baby and the surrounding environment will be acquired with a low-voltage sound detection circuit utilizing an electret microphone and the versatile LM386 op-amp, as shown below in Figure 3.2. The design decision to create a sound detection circuit opposed to using a sound sensor module was based on the ability to add filtering components to the circuit.

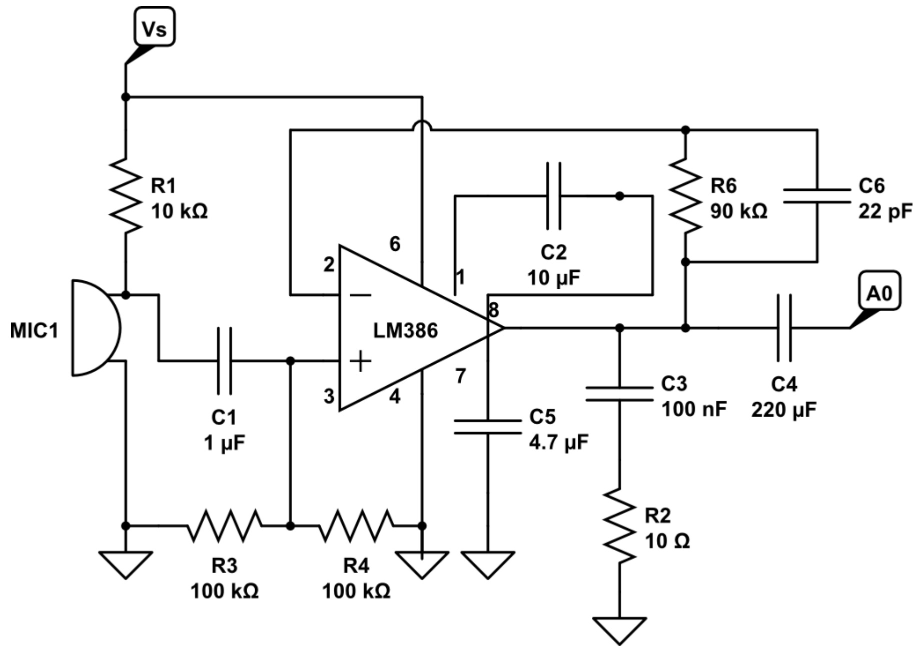

Figure 3.2: Sound Sensor Circuit

The sensor circuit designed for the POC meets the design requirements described in Table (3.2).

Table 3.2: Sound Sensor Circuit Specification

	Requirement Code	**Design Specification**
D3.2.1	R3.2.1-POC-2	Electret microphone shall have a sensitivity range of 40Hz to 20kHz.
D3.2.2	R3.2.1-POC-2	Electret microphone shall have a gain of 200.
D3.2.3	R3.2.1-POC-2	Sound pressure level (dB) should have a minimum listening distance of 28cm.
D3.2.3	R3.2.1-POC-2	Sound pressure level should be accurate within ± 1 dB from 28cm away.

The LM386 is an 8-pin operational amplifier (op-amp) that allows for a voltage gain between 20-200 [7]. The gain of the circuit defines a range of possible volume levels and is adjusted with a capacitor.

In order to sample a range of volume levels, a gain = 200 was achieved with a C_2 = 10uF across pin(1) and pin(8) as shown in the circuit in Figure 3.2.

An electret microphone is chosen for it's built-in time-varying capacitance which buffers and amplifies the signal [8]. The chosen electret microphone is biased and connected to the 5V source and has a frequency range of 40Hz-13KHz. Its load resistor sets a low impedance which will be followed by the high impedance amplifier circuit.

The selected cut-off frequency of the filter is chosen to place the cut-off frequency 25% above or below the desired low- or high-pass filter frequency to achieve ~0.25dB reduction at the desired frequency [9]. The cut-off frequency of 80kHz was calculated using (6).

$$f_{cutoff} = \frac{1}{2pi * R_6 C_6} \quad \textbf{(6)}$$

In order to reduce sampling noise, decoupling capacitors are added at the nodes after the microphone as well as at the output of the op-amp to filter any DC noise coupled with the signals after sampling and amplification. A high pass filter utilizing a parallel combination of resistors is placed at the non-inverting input providing a high frequency path of 3.18Hz, using (7). Implementation of these filters provides a wide frequency range for detection where everyday noises and human-made noises are subsets of.

$$f_{highpass} = \frac{1}{(2pi * R_3 || R_4 * C_1)} \quad \textbf{(7)}$$

Lastly, utilizing the 12-bit ADC of the ESP32, the sound sensor is able to detect a wide range of sound intensities within the room, which should allow for loud ambient noises in the infant's environment and for the infant's cries to be detected. The subsystem circuit overall should be able to reliably detect sounds within the room.

3.3. Position Identification

In order to determine a baby's position, sensors are required to gather positional data. This will be done through a sensor matrix mat, utilizing a piezoresistive material: Velostat. Velostat is a material made of a polymer foil injected with carbon black to make it conductive [10]. Upon applying pressure to this material, the resistance in the corresponding area decreases, proportional to the amount of pressure applied. By sandwiching Velostat between columns and rows of copper tape, sensors at each intersection are formed. A vinyl sheet on the top and bottom layers insulate the exposed conductive layers. Figure 3.3 shows the mat and each of its layers. Given Velostat's low cost, sheet form-factor, and pressure-sensing capabilities, it was the obvious choice for our implementation.

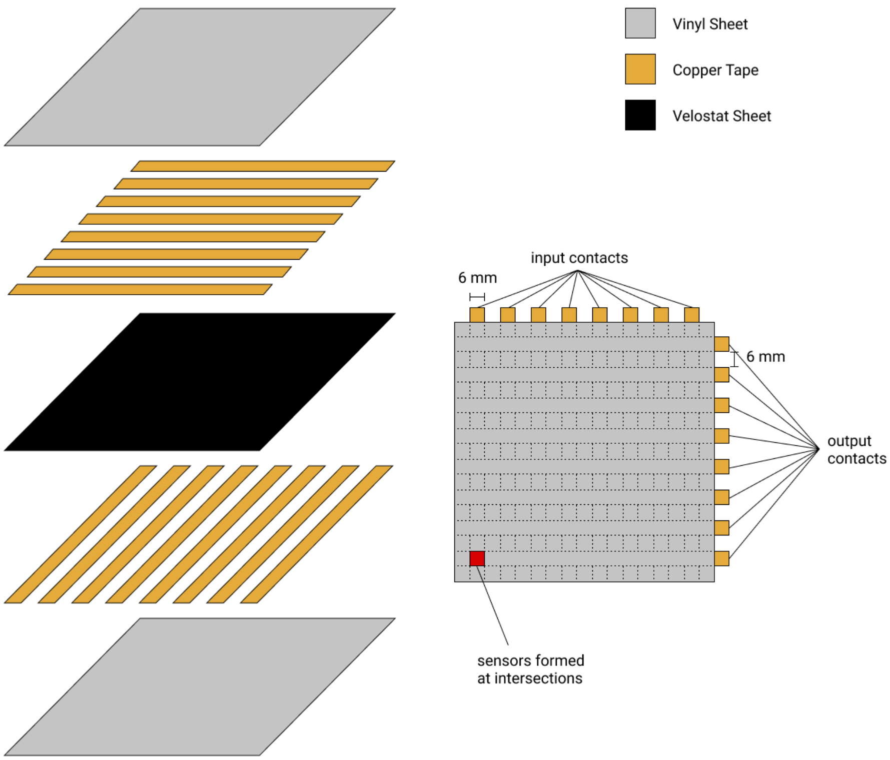

Figure 3.3: Pressure Mat Layers and Aerial View

3.3.1 Sensor Size and Spacing Approximations

The size and spacing of the sensors must be chosen for sufficient resolution, as well as a high enough density to detect changes in pressure. The latter is what will be focused on as the bottle-neck to the design, and is determined by Velostat's dynamic range. Velostat's dynamic range is the range of applied pressure that induces a change in resistance. Based on the experimental data seen in Figure 3.4, the dynamic range of Velostat is around 0 to 1kg. If a baby's weight distribution exceeds 1kg per sensor, changes in position would not be observed.

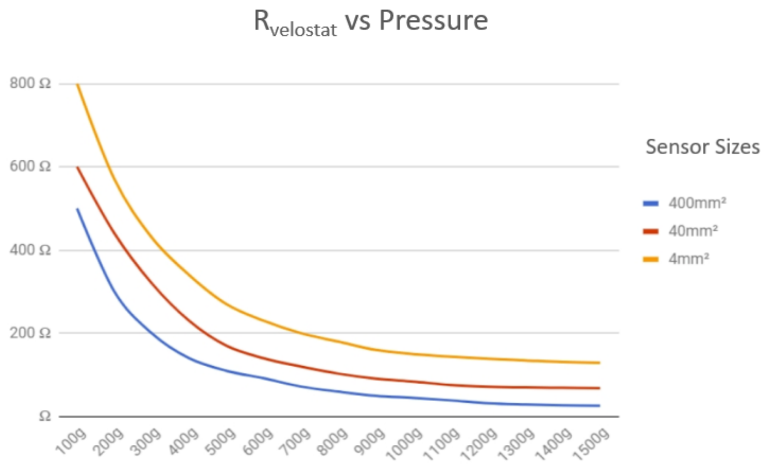

Figure 3.4: Experimental Data - Velostat Resistance vs. Pressure [11]

The sensor matrix must be designed to avoid this. As shown previously in Figure 3.3, the sensor sizes chosen are $6mm^2$ with 6mm gaps, based on copper tape's 6mm width. With this in mind, we will confirm that such a choice can support a baby's weight well within Velostat's 1kg dynamic range.

With limited data on specific baby dimensions and weights, approximations will be used based on the largest baby the Tenshi Crib is rated for: a 6-month old baby boy. A 6 month-old baby boy weighs on average 7.9kg, and is approximately 67.6cm long [12]. Shoulder to shoulder measurements are approximately 19.5cm [13]. For simplicity and to leave room for larger babies, a weight of 9kg, length of 70cm and shoulder-to-shoulder width of 20cm is chosen. Using the length and shoulder-to-shoulder width, an upper-boundary of the cross-sectional area a 6-month old baby could occupy is found (8):

$$area = width\ x\ length$$

$$= 70\ cm\ \times 20\ cm$$
$$= 1400\ cm^2 \quad \textbf{(8)}$$

With this, converting to mm, the average weight distributed is shown in (9):

$$weight\ distribution = \frac{weight}{area} \quad \textbf{(9)}$$

$$= \frac{9\ kg}{140000\ mm^2}$$

$$= 0.0000642857\ \frac{kg}{mm^2}$$

Multiplying by our sensor area in (10):

$$pressure\ felt\ by\ one\ sensor = weight\ distribution\ x\ sensor\ area \qquad \textbf{(10)}$$

$$= 0.0000642857\ \frac{kg}{mm^2} \times 6\ mm^2$$

$$= 0.0003857\ kg$$

The average pressure a single sensor may detect in this use case is found. As we know, a baby's weight is not distributed evenly throughout a rectangle. However, with such a low value, it is safe to assume that a baby's weight distribution will be well within Velostat's 1kg dynamic range.

3.3.2 Circuit Layout

With the sensors layed out in mat form, a way to sample each of the points in the matrix is needed. The columns of the copper tape on the bottom layer, and the rows on the top layer will each extend past the Velostat on one side each, enabling contacts to be created. The 5V output of the ESP32 will connect to an analog demultiplexer, which will then branch out to each of the column contacts. This way, a voltage can be applied to a given column based on the address fed to the multiplexer through the ESP32's GPIO pins. Similarly, the rows, which represent the output, will be connected to a fixed resistor to ground, and branch to an analog multiplexer, which is connected to an ADC pin in the ESP32. This forms a voltage divider with the Velostat acting as a variable resistor. Figure 3.5 shows the circuit described.

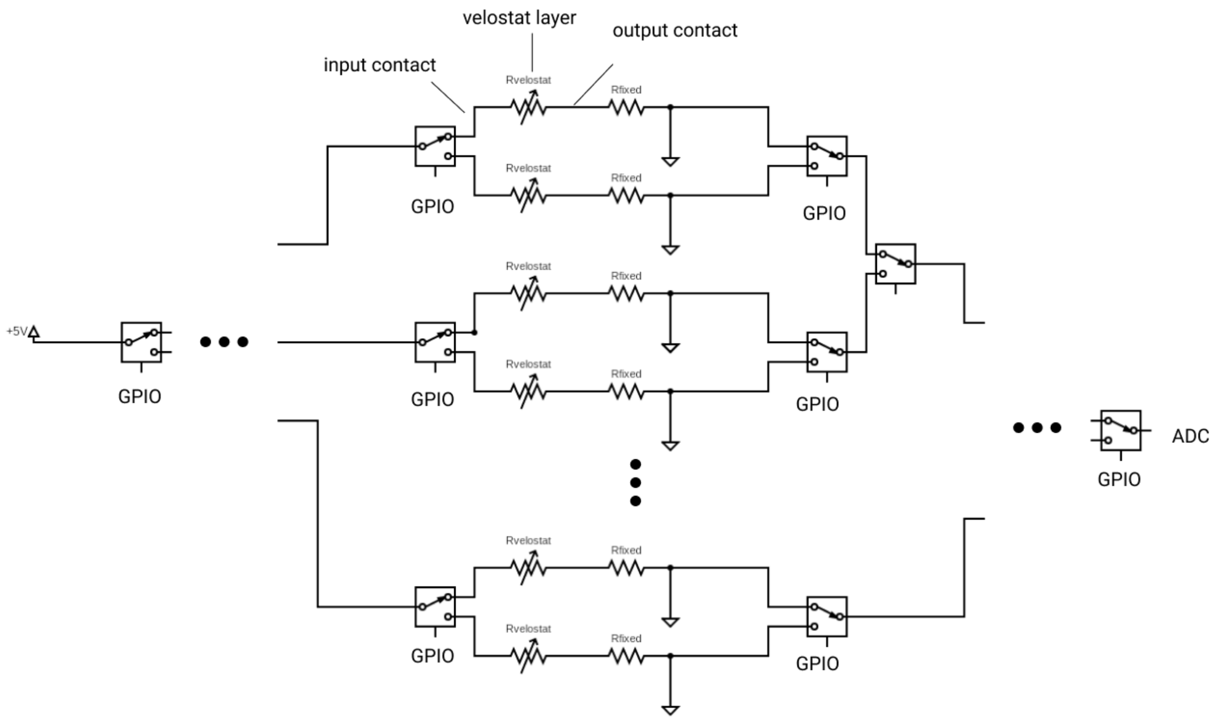

Figure 3.5: Circuit Diagram of Pressure Mat for n Number of Sensors

3.3.3 Choice of R_{fixed}

Given the following Equation (11) for a voltage divider:

$$V_{out} = V_s \frac{R_{Velostat}}{R_{Velostat} + R_{fixed}} \tag{11}$$

The choice of R_{fixed} determines the sensitivity of the output to the change in $R_{velostat}$. Therefore, the sensitivity should be maximized such that the slightest amount of pressure can be seen. From previously shown Figure 3.4, it can be seen that $R_{velostat}$ for small sensor sizes range from 0 to 500 Ω. The graph in Figure 3.6 shows the voltage divider equation for varying values of R_{fixed} within that range.

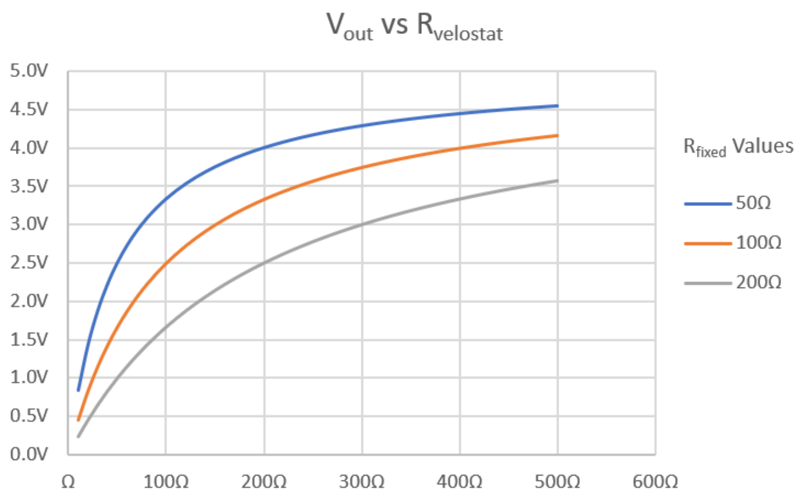

Figure 3.6: Voltage Divider Equation for Varying R_{fixed}

The value chosen for R_{fixed} is 200Ω. This is because a greater sensitivity at higher values of $R_{velostat}$, portrayed by the slightly steeper slope on the graph, is desired. The trade-off is the smaller voltage range that can be covered (0 - 3.5V vs 0 - 4.5V), which offers ample room to work with.

3.3.4 Pseudo-code

With the full circuit layout, sampling the full matrix is trivial. In the Arduino IDE, looping through the input and output mux address values and reading the analog output achieves this. The following pseudo-code is described:

```
while(ON) {
      pressure_values = [][]; // 2D matrix
      input_mux_select = 0;
      // loop through all input/output mux values
```

```
for column in pressure_values{
      set_mux_address(input_mux, input_mux_select);
      for row in column{
            set_mux_address(input_mux, input_mux_select);
            row = readAnalog(); // read sensor voltage val
            output_mux_select += 1;
            }
      input_mux_select += 1;
      }
}
```

The full matrix is now capable of being sampled for the location and degree of pressure being applied. This data is utilized by the Data Analysis System's machine learning algorithm to determine baby position. More in depth discussion of this algorithm can be found in section 4.

3.3.5 Pressure Sensor Design Specs

Table 3.3: Pressure Sensors Specification

	Requirement Code	Design Specification
D.3.3.5.1	R3.2.1-POC-3	The material Velostat shall be used which has a dynamic range of 0 to 1kg on a given sensor.
D.3.3.5.2	R3.2.2-POC-2	Every full scan of the pressure mat shall have an attached timestamp.
D.3.3.5.3	R3.3.1.2-POC-2	The pressure mat shall use a dedicated core on the ESP32 to sample each sensor.

4. Data Analysis System

4.1 Data Processing

4.1.1 ESP32-WROOM-32

Acting as the main compute unit for the Data Analysis system, the ESP32-WROOM-32 best fits the application of the Tenshi Baby Crib's features. This section will introduce the ESP32 and how it is used to implement the Data Analysis system.

The ESP32 is a powerful microcontroller that can operate at different clock speeds. Given its dual core 32-bit LX6 microprocessor, the ESP32 outperforms most of the microcontrollers in its class. This is necessary to sample each of Tenshi Baby Cribs' sensor systems at a sufficient rate. Additionally, the microcontroller has the built in ability to connect through Wi-Fi, essential for connecting to the Tenshi Baby Crib mobile app. In short, the data analysis system will make use of its Wi-Fi capabilities, its powerful dual core processing, and analog-to-digital converter, as seen in Figure 4.1 below.

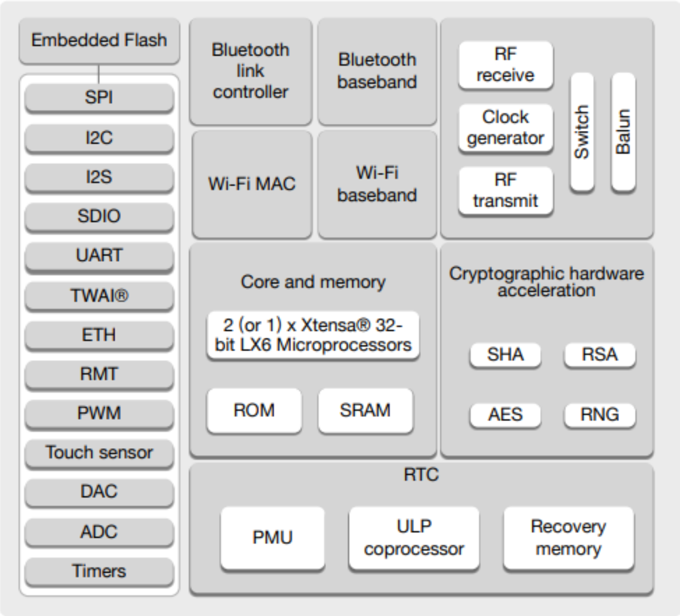

Figure 4.1: ESP32 Block Diagram [14]

On the ESP32 are multiple general purpose input/output pins that will interface with the Sensor System. These serve as the inputs to the processing algorithms that will run on the microcontroller, as seen in Figure 4.2 below. These computation algorithms will be further detailed in later sections.

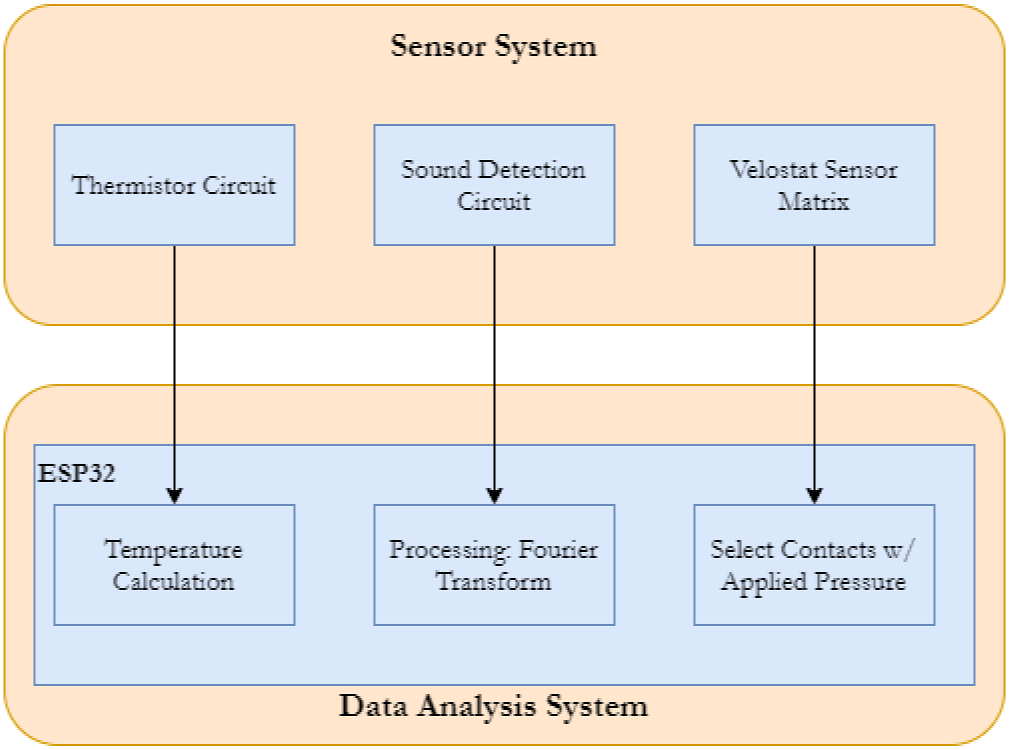

Figure 4.2: Sensor and Data Analysis System Interaction

Once the inputs have been processed by the ESP32, it will utilize its built in Wi-Fi capabilities to interface with the Notification System. The ESP32 will act as a Wi-Fi station connected to a router, which allows other devices such as smartphones to communicate with the microcontroller on the same network. This means of communication will be crucial to implementing the Notification System, detailed in Section 5.1.

4.1.2 Multiprocessing

The dual core microprocessor is one of the key factors this microcontroller was selected for the project. Multiprocessing enables the Tenshi Baby Crib to frequently sample and process input data efficiently from the various Sensor System components. For this design, one core will be allocated for simply computing temperature and processing signals from the sound detection circuit. The other core will be allocated for the pressure grid, sampling multiple pressure contacts and their applied pressure values. The implementation shown in Figure 4.4 will allow the design to meet the requirement specification of sampling data every second.

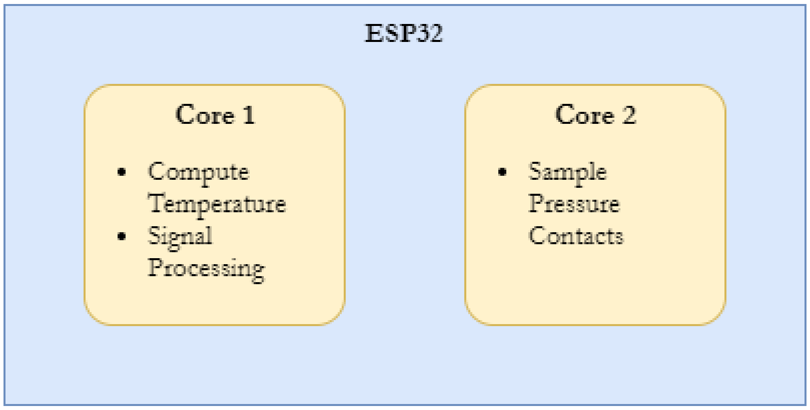

Figure 4.3: Multiprocessing with Two Cores

4.1.3 Powering the ESP32

The ESP32's ideal operating voltage is 3.3V, which can be supplied through micro-usb, a 3.3V battery, and a regulated or unregulated external power supply [15]. For the system's purposes, the ESP32 will be powered by connecting the micro-usb port on the microcontroller to a power supply [16]. Optimizing power efficiency of the microcontroller is difficult due to constant sensor sampling. When connected to wifi, the ESP32 is in the "active mode" and the power consumption can range from 80mA to 260mA [17]. For reference, when the microcontroller is in "modem sleep mode" which is when the wifi, bluetooth, and radio features are disabled, the power consumption ranges from 3mA to 20mA. With wifi connection being a key factor, the ESP32 will always be in active mode to track sensor data to send to the mobile phone.

4.1.4 ESP32 Design Specs

When reviewing the specs, compatibility and functionality of the ESP32, the microcontroller presents the capabilities of operating as our project's data analysis system. Furthermore, the ESP32 addresses multiple requirements from the requirement specification document. All ESP32 design specifications will be in regards to at least one requirement and are listed in Table x below.

Table 4.1: Microcontroller Design Specification

	Requirement Code	Design Specification
D.4.1.5.1	R3.2.2-POC-1, R3.2.2-POC-3, R3.2.2-POC-4, R3.2.2-PT-6	No more than 34 GPIOs shall be used.
D.4.1.5.2	R3.3.1.2-POC-2	The microcontroller shall be able to multiprocess.
D.4.1.5.3	R3.3.1.2-PT-3	The microcontroller shall not use an external wifi module.
D.4.1.5.4	R3.3.1.2-PT-3	The microcontroller shall indicate a failed connection with the notification system if wifi is not connected.
D.4.1.5.5	R4.1-PT-2	The power supply for the microcontroller can range from 3V to 3.6V.

4.2. Temperature Monitoring Pseudo-code

From the thermistor circuit, temperature can be calculated into Celsius (°C) with the ESP32 as described in Section 3.2. Once the temperature is calculated, the simple vulnerability detection software within the microcontroller will flag when temperature is less than 20°C or greater than 23°C. To suppress acute malfunctions in the circuit, the measurements will take a rolling average of the measurements in the past minute. As such, the following pseudo-code implements the intended temperature detection algorithm.

```
vulnerableFlag = false;
tempBuffer[60]; //circular buffer
while(ON) {
      //measure from circuit
      currTemperature = measureTemperature();
      //update circular buffer - moving pointers, etc.
      updateCircularBuffer(tempBuffer, currTemperature);
      //average elements in buffer
      temperature = average(tempBuffer);
      // < 20C or 23C <
      updateFlags(temperature, vulnerableFlag);
      if(coldFlag || hotFlag) {
            //notification system interactions
      } //if
} //while
```

The circular buffer containing the past minute of samples will allow for efficient execution in the ESP32. Details of the notification system interaction will be further detailed in Section 5 and more during the prototype phase of the project.

4.3. Sound Detection Pseudo-code

In order to detect loud noises within the room and cries of the baby the sound sensor data utilizes the ESP32's 12-bit ADC. Using a sampling frequency 2048 and 128 number of samples, sound pressure (dB) calculated and frequency of the sound can be observed.

To prevent false positives, the sampled data is averaged to verify the legitimacy of sound and a dynamically adjusting threshold is implemented. The approach of using relative increase to compare the momentary difference against the threshold accounts for occasional shifts in the sensor's baseline.

The fast Fourier Transform (FFT) of the analog input provides the discrete Fourier transform. The algorithm will utilize this Fourier analysis to convert the signal into the frequency domain for detection. Alongside FFT, another means of getting sound information is to calculate the decibel of the sound is calculated, as shown in (11).

$$dB = 20\log\left(\frac{ADC}{ADC_{ref}}\right) + ADC_{ref} \tag{11}$$

The value of ADC_{ref} is the voltage reading at quiet room levels as read by ESP32 and is used to calibrate the microphone. Additionally, considering that the human ear is more sensitive to higher sound pressures in the 1 to 4kHz range, it is beneficial to compare the found values to dB(A) [18]. The value of dB(A) is approximately the inverse of 40dB (at 1kHz) [19].

Using this data, the following pseudo-code implements the intended sound detection algorithm:

```
detectionFlag = false;
samples = 128;
samplingFrequency = 2048;
soundHzBuffer[samples];
soundDecibelBuffer[samples];

while(ON) {
      for(int i=0; i<samples; i++){
            soundHzBuffer[i] = analogRead(A0);
            soundDecibelBuffer[i] = toDecibel(A0);
      }
```

```
        //finds peak frequency amongst samples
        peak = findPeak(soundBuffer, SAMPLES);
        dB = average(soundDecibelBuffer);
        dBA = toDecibelA(dB);
        //sets new peak frequency
        setThreshold();
        //tracks how long the peak as been set
        //(if dB >= 60dB && freq = constant > 30 mins)
        //|| (if dB >= 80dB && freq = constant)
        vulnerableFlag = compareThreshold(peak, dB, dBA);
        updateFlags(temperature, vulnerableFlag);
        if(cryFlag || roomNoiseFlag) {
              //notification system interactions
        } //if
} //while
```

With the above analysis of the sound sensor data, detection flags can be set and sent to the notification system.

4.4. Position Identification Implementation

Taking input from the pressure grid via the ESP32, the Data Analysis system will implement a classification machine learning algorithm to identify whether the baby is on their back or front. The model will be trained using a supervised learning approach. During development, designers will collect training data with dolls of various weights, mapping extracted inputs to outputs labelled as ‘front’ or ‘back’. As such, the supervised learning approach to training the machine learning model is ideal for classification [20]. The depiction of the approach is shown in Figure 4.5.

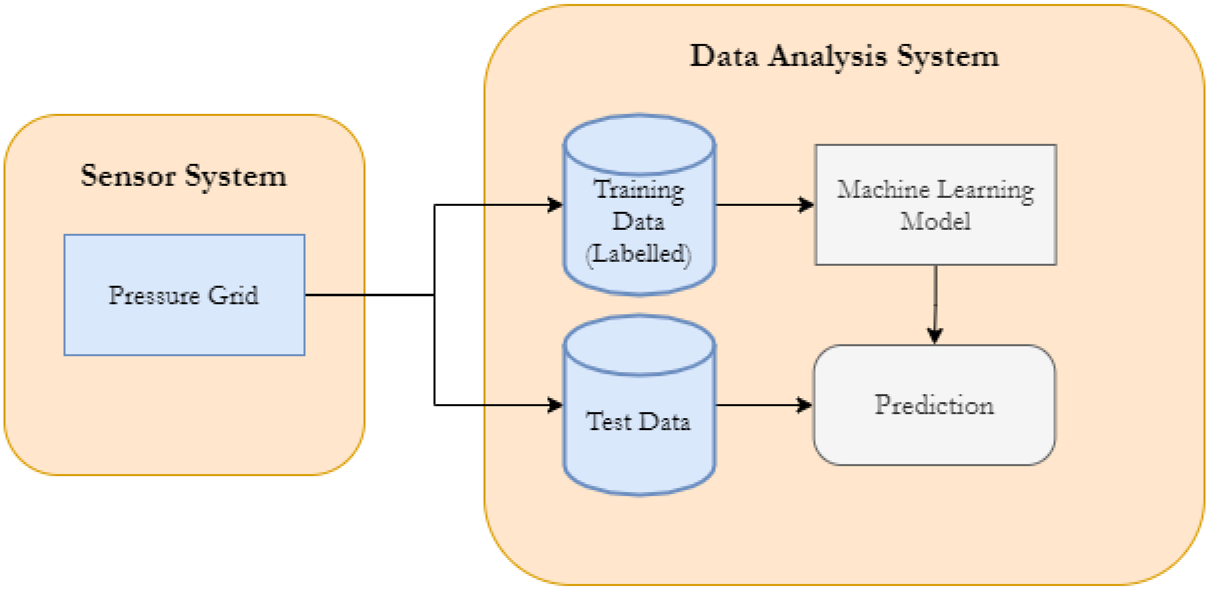

Figure 4.4: Supervised Learning

The algorithm at which the machine learning model uses to identify the position of the baby is undetermined during this time. However, the inputs that will be used as features to predict the baby’s position are known. These inputs are as follows in Table 4.2 below.

Table 4.2: Machine Learning Model Inputs

Pressure Grid Input	**Value**
Location of “closed” contacts	2D Discrete Position (x,y)
The amount pressure applied per contact	Voltage [0V, 3.5V]

In short, these inputs form 2D matrices with varying applied pressure values. Creating a more granular distinction of these inputs is possible. Taking those matrices the model can be trained by implementing one of a number of common machine learning algorithms. Namely, Support Vector Machines, Decision Trees, and Deep Learning. Each of these algorithms have their own strengths and weaknesses which are outlined below in Table 4.3.

Table 4.3: Machine Learning Algorithms Comparison [20]

ML Algorithm	Strengths	Weaknesses
Support Vector Machines	Model non-linear decision boundaries and robust to overfitting in multiple dimensions.	Memory intensive, does not scale to larger datasets.
Decision Tree	Robust to outliers and non-linear relationships.	Prone to overfitting data.
Deep Learning	Widely used for classification of audio, text, and image data.	Large amount of data required to train.

Performing initial tests with widely available machine learning libraries such as PyTorch, TensorFlow will allow us to determine the ideal machine learning algorithm for classification. If the 2D matrices do not offer the accuracy needed to classify the baby's position, we can further distinguish and define the data into more features, such as time and averaging. To aid in our decision, the implementation of our classification model must achieve these design specifications in Table 4.4 so as to meet requirements.

Table 4.4: Position Data Analysis Specifications

	Requirement Code	Design Specification
D4.4.1	R3.2.2-PT-6 R3.2.2-PT-7	Machine Learning Model will be able to identify baby positions with 95% accuracy.
D4.4.2	R3.2.2-PT-11 R3.3.1.1-PT-1	Machine Learning Model implementation will be optimized to be space and time efficient to determine vulnerabilities in less than 5 seconds.

5. Notification System

5.1. Interfacing with the Data Analysis System

Once a connection to the network has been established, the Notification System will use an API to manage the data being sent and retrieved, then using a database to store the data. This relationship is shown in Figure 5.1 and detailed in this section.

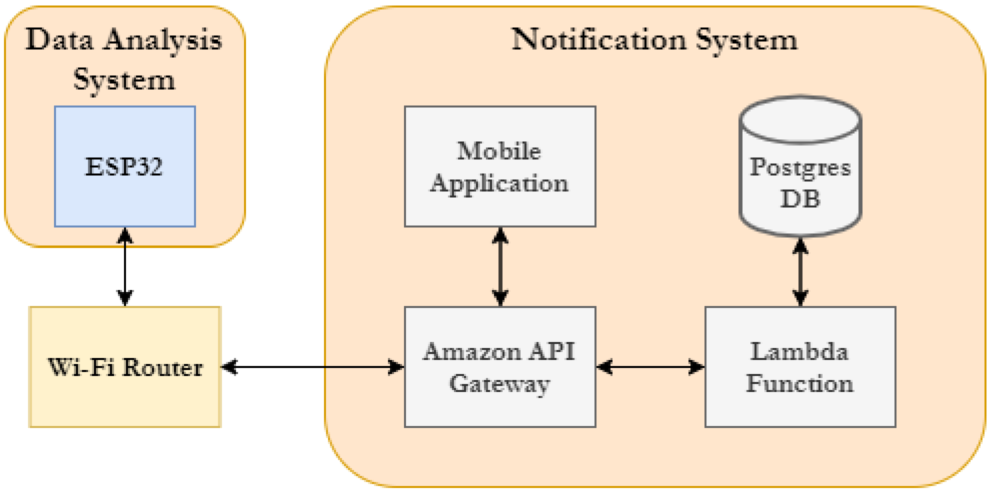

Figure 5.1: Data Analysis and Notification System Interaction

5.2. Mobile Application

The mobile application will be created with the React Native framework. By choosing React Native, this framework can develop an app for both android or apple devices. Picking a framework that would only support one of the options would greatly divide the target market.

The users will be able to create an account, login, and pair their app to their specific Tenshi Baby Crib model. The API as well as the database will work in tandem to authorize and check if the login credentials are valid. A strong authorization algorithm is important for user security.

As previously seen in Figure 5.1, the data analysis system works in tandem with the notification system. A wifi router, API, and a database is what is needed for these two systems to connect with each other. Aside from cross platform development, React Native was chosen because of its huge development community. There will be multiple resources online to help when integrating the mobile app with the ESP32. Additionally, the notification system is targeted for the prototype phase and will not be fully completed by the proof of concept phase. Since React Native is a new framework for the entire Tenshi team, more time is needed to create a well designed mobile app.

5.3. Creating An API

An API will be used to connect the data from the data analysis system to the notification system. The API will be created through Amazon Web Services (AWS). The Amazon API gateway feature of AWS will be used to construct the API. The API facilitates data transfer between the system by using HTTP methods such as "GET" and "POST". Consequently, an AWS lambda function will be created to help transfer information to the API as cron jobs and for finding unique sets of data; a critical feature for Tenshi's application.

5.4. Selecting A Database

The notification system only reads and stores data, therefore, most database systems should be sufficient. PostgreSQL will be the relational database management system (RDBMS) database used to store analyzed information from the sensor systems after considering the company's skills. Moreover, PostgreSQL has a native feature which allows JSON data to be parsed easily which will better facilitate saving and sending data functionalities.

5.5. Data Visualization

The mobile application will include graphs to visualize the sampled sensor data. To present the data in an efficient, intuitive and informative manner, Victory Native will be used for it's compatibility with React Native and comprehensive documentation.

5.6 Notification Design Specs

There are multiple components that make up the connection with the data analysis system as well as the notification system. Each component plays a key role to create a functional system. All the design specifications for the notification system will be listed in Table 5.1 below.

Table 5.1: Mobile Application Design Specification

	Requirement Code	Design Specification
D.5.6.1	R3.2.2-PT-8	The mobile app can graph in different date ranges.
D.5.6.2	R3.2.2-PT-9, R3.2.2-PT-10, R3.3.1.1-PT-1	The notification system shall be able to receive data and alert users when the baby is vulnerable.
D.5.6.3	R3.2.2-PT-12	The mobile app and ESP32 should be connected to the same wifi router for interaction.
D.5.6.4	R3.3.1.3-PT-1, R3.3.1.3-PT-2	The authorization algorithm shall be robust and prevent fake accounts to receive other users data.
D.5.6.5	R3.3.1.3-FP-3	The data received should be securely stored in the database.
D.5.6.6	R3.3.1.4-FP-4	The mobile app will be accessible for android and apple users.
D.5.6.7	R3.2.2-PT-8	The API and lambda function will be able to run cron jobs for daily and weekly updates.
D.5.6.8	R3.3.1.4-PT-2	The graphed data is intuitive to read and understand.

6. Conclusion

The goal of the Tenshi Baby Crib is to provide smart technology for caregivers of infants to informatively and efficiently monitor their baby and their environment. The aforementioned design specifications are to outline the comprehensive design consideration taken into account to build the Sensor System and the Data Analysis System for the PoC of the Tenshi Baby Crib.

Within the Sensor System, temperature, sound and position have been specified to meet their respective requirements specifications in the PoC phase. It must be able to measure temperature room temperature ranges of under 16°C to over 22°C. This will be met by implementing a temperature circuit utilizing a 10K NTC thermistor. In addition, the sound circuit ,consisting of an electret microphone and an LM386 op-amp must be able to measure noise levels of 40 dB(A) to 120db(A). Most of all, the pressure grid consisting of a pressure matrix utilizing Velostat must be built and primed to extract pressure data ranging from 0kg to 1kg.

To deliver information to our target users, Tenshi has progressed towards a functioning Data Analysis and Notification System. The ESP32 system-on-chip microcontroller will be used to handle the sensor systems' data and facilitate communication between the sensor systems and the mobile application. The ESP32 was chosen specifically for its performance, its 12-bit ADC and its on-board Wi-Fi module. The Data Analysis System's ability to log temperature and sound measurements has been specified in their respective sampling and averaging algorithms. Lastly, with the design choice to use React Native, the Data Analysis System shall be able to visual datasets for caregivers with compatible chart libraries and machine learning. In order to manage the Sensor System data, the PostgreSQL RDBMS will be part of the system's architecture for it's compatibility with React Native. In conclusion, the PoC phase of the Tenshi Baby Crib will include the basic functionality of the Sensor System and the Data Analysis System, while preparing for the future Prototype phase, implementing the Notification System and optimized algorithms to facilitate healthy growth for infants.

7. References

[1] "Is Your Child Safe? Sleep Time" Canada.ca. [Online]. Accessed: Feb. 21, 2021. Available: https://www.canada.ca/en/health-canada/services/consumer-product-safety/reports-publications/consumer-education/your-child-safe/sleep-time.html.

[2] "Humidity and Temperature for Sleeping Babies and Children: Heartistic Birthing Doula," Nourish Birth + Postpartum, Jul. 2020. [Online]. Accessed: Feb. 21, 2021. Available: https://nourishbirthpostpartum.com/humidity-and-temperature-for-sleeping-babies-and-children/.

[3] Amazon. "uxcell 10K NTC Thermistor Probe 19.7 Inch Stainless Steel Sensitive Temperature Temp Sensor for Air Conditioner: Home Improvement." Amazon.com. Accessed: Mar. 21, 2021. [Online]. Available: https://www.amazon.com/uxcell-Thermistor-Stainless-Temperature-Conditioner/dp/B07LGL6TQM#productDetails

[4] E-Tinkers. "Using a Thermistor with Arduino and Unexpected ESP32 ADC Non-linearity." e-tinkers.com. Accessed: Mar 21, 2021. [Online]. Available: https://www.e-tinkers.com/2019/10/using-a-thermistor-with-arduino-and-unexpected-esp32-adc-non-linearity/

[5] Ametherm. "NTC Thermistors and Hart Equation." ametherm.com. Accessed: Mar 21, 2021. [Online]. Available: https://www.ametherm.com/thermistor/ntc-thermistors-steinhart-and-hart-equation

[6] P. Kane. "Thermistors/Temperature Measurement with NTC Thermistors." Jameco.com. Accessed: Mar 21, 2021. [Online]. Available: https://www.ametherm.com/thermistor/ntc-thermistors-steinhart-and-hart-equation

[7] Texas Instruments. "LM386 Low Voltage Audio Power Amplifier." ti.com. Accessed: Mar 22, 2021. [Online]. Available: https://www.ti.com/lit/ds/symlink/lm386.pdf

[8] B. York. *Microphone Circuit.* Accessed: Mar. 22, 2021. [Online]. Available: https://web.ece.ucsb.edu/Faculty/rodwell/Classes/ece2c/labs/Lab1b-2C2007.pdf

[9] B. B. Monson, J. Caravello. "The maximum audible low-pass cutoff frequency for speech," *The Journal of the Acoustical Society of America,* vol. 146, no. 6, Dec. 2019. Accessed on: Mar. 22, 2021. [Online]. Available: https://doi.org/10.1121/1.5140032

[10] P. J. Zilinskas, T. Lozovski, J. Jurksus. "Electrostatic Properties and Characterization of Specific Polymeric Materials for Building Purposes," *Materials Science (Medziagotyra),* vol. 16, no. 1, Jan. 2010. Accessed on: Mar. 22, 2021. [Online]. Available: https://www.researchgate.net/publication/264877580_Electrostatic_Properties_and_Characterization_of_Specific_Polymeric_Materials_for_Building_Purposes

[11] M. Reps. "Pressure Sensor Matrix Mat Project." reps.cc. Accessed: Feb. 20, 2021. [Online]. Available: https://reps.cc/?p=50

[12] D. Murray. "Average Baby Weight and Length During the First Year." verywellfamily.com. Accessed: Mar. 20, 2021. [Online]. Available: https://www.verywellfamily.com/first-year-infant-growth-431721

[13] Craft Yarn Council. "Baby Size Chart." craftyarncouncil.com. Accessed: Mar. 15, 2021. [Online].
Available: https://www.craftyarncouncil.com/standards/baby-size-chart

[14] Espressif. "ESP32 Series." espressif.com. Accessed: Mar. 10, 2021. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

[15] Tech Explorations. "ESP32 dev kit power options." techexplorations.com. Accessed: Mar. 25, 2021. [Online]. Available: https://techexplorations.com/guides/esp32/begin/power/

[16] How To Electronics. "Power Supply for ESP32 with Battery Charger & Boost Converter." how2electronics.com. Accessed: Mar. 25, 2021. [Online]. Available: https://how2electronics.com/power-supply-for-esp32-with-boost-converter-battery-charger/

[17] Last Minute Engineers. "Insight Into ESP32 Sleep Modes & Their Power Consumption." lastminuteengineers.com. Accessed: Mar. 25, 2021. [Online]. Available: https://lastminuteengineers.com/esp32-sleep-modes-power-consumption/

[18] IAC Acoustics. "Noise Control Concepts." iacacoustics.com. Accessed: Mar. 25, 2021. [Online]. Available: https://www.iacacoustics.com/blog-full/noise-control-concepts.html

[19] The Engineering ToolBox. "Decibel A, B, and C." engineeringtoolbox.com. Accessed: Mar. 25, 2021. [Online]. Available: https://www.engineeringtoolbox.com/decibel-d_59.html

[20] Elite Data Science. "Modern Machine Learning Algorithms: Strengths and Weaknesses." Accessed: Mar. 24, 2021. [Online]. Available: https://elitedatascience.com/machine-learning-algorithms

[21] D. A. Norman, *The Design of Everyday Things,* Revised and Expanded ed. New York, New York, USA: Basic Books, 2013.

[22] Interaction Design Foundation. “Make it Easy on the User: Designing for Discoverability within
Mobile Apps.” Interaction-design.org. Accessed: Mar 20, 2021. [Online]. Available: https://www.interaction-design.org/literature/article/make-it-easy-on-the-user -designing-for-discoverability-within-mobile-apps

[23] *Systems and software engineering - Software life cycle processes,* ISO/IEC/IEEE 12207-2017, Nov. 2017. Accessed: Mar. 18, 2021. [Online]. Available: https://www.iso.org/standard/63712.html

[24] *Information technology - User interface icons - Part 1: Introduction to and overview of icon standards,* ISO/IEC TR 11581-1:2011, Dec. 2011. Accessed: Mar. 18, 2021. [Online]. Available: https://www.iso.org/standard/46444.html

[25] *Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 1: General requirements,* IEC 61508-1:2010, Apr. 2010. Accessed: Mar. 18, 2021. [Online]. Available: https://webstore.iec.ch/publication/5515

[26] *Ergonomics of human-system interaction - Part 161: Guidance on visual user-interface elements,* ISO 9421-161:2016, Feb. 2016. Accessed: Mar. 18, 2021. [Online]. Available: https://www.iso.org/standard/60476.html

[27] *Acoustics - Measurement of airborne noise emitted by information technology and telecommunications equipment,* ISO 7779:2018, Nov. 2018. Accessed: Mar. 18, 2021. [Online]. Available: https://www.iso.org/standard/69857.html

[28] *General requirements for battery-powered appliances,* UL 2595, Sep. 2015. Accessed: Mar. 18, 2021. [Online]. Available: https://standardscatalog.ul.com/ProductDetail.aspx?productId =UL2595

[29] *Cribs, Cradles and Bassinets Regulations,* SOR/2016-152, Dec. 2016. Accessed: Mar. 18, 2021. [Online]. Available: https://laws-lois.justice.gc.ca/eng/regulations/SOR-2016-152

[30] *Information technology equipment - Safety - Part 1: General requirements,* IEC 60950-1:2005, Dec. 2005. Accessed: Mar. 18, 2021. [Online]. Available: https://webstore.iec.ch/publication/4020

[31] *Semiconductor devices - Micro-electromechanical devices - Part 33: MEMS piezoresistive pressure-sensitive device,* IEC 62047-33:2019, Apr. 2019. Accessed: Mar. 18, 2021. [Online]. Available: https://webstore.iec.ch/publication/31615

[32] Apple Inc. "Human Interface Guidelines." Apple.com. Accessed: Mar. 20, 2021. [Online]. Available: https://developer.apple.com/design/human-interface-guidelines

[33] Google Developers. "Core app quality." Android.com. Accessed: Mar. 20, 2021. [Online]. Available: https://developer.android.com/docs/quality-guidelines/core-app-quality

[34] H. H. Tsang. (2019). CMPT 275: Software Engineering I - Mobile User Interface Design [PowerPoint slides]. Available: https://canvas.sfu.ca/courses/47032/files

[35] M. Sjoerdsma, W. C. Scratchley. (2021). Human Factors & Usability Engineering [PowerPoint slides]. Available: https://canvas.sfu.ca/courses/60611/files

[36] Droids on Roids. "Flutter vs. React Native - What to Choose in 2021?" thedroidsonroids.com. Accessed: Mar. 25, 2021. [Online]. Available: https://www.thedroidsonroids.com/blog/flutter-vs-react-native-what-to-choose-in-2021

Appendix A: User Interface

A.1. Introduction

The user interface (UI) and appearance design appendix documents the analysis, adherence to standards, and planned tests pertaining to the creation of the user interactive elements for the Tenshi Baby Crib. Majority of these are intended to be implemented in the prototype phase of the project as the company focuses on the Sensor System and Data Analysis System electronics for the proof of concept. Regardless, by following the items outlined in this document, all interactive elements of the design will enable the end users to utilize all features of the product in an efficient and comfortable fashion.

This appendix will first outline the required knowledge of end users and take into consideration any restrictions poised by our product. Then it will analyze human factors through seven elements of user interface interaction in order to further specify the design for effective human use. In addition, this document lists user interface and design engineering standards to ensure consistency and compliance. Finally, it will propose analytical and empirical usability testing procedures that will be performed to verify and validate that users can utilize all features offered by the Tenshi Baby Crib.

A.2. User Analysis

The system is designed to safely and reliably monitor an infant from a distance to alleviate the responsibility of a busy caregiver. The system should be integrated into their habitual tasks and not disturb the infant. The user interface is designed to ensure the caregiver is sufficiently informed about the system to utilize all of Tenshi baby crib's functionalities.

The Tenshi baby crib system will have an instructional set-up guide for all its components, therefore specific prior knowledge of cribs or smart technology is not necessary for use. Given the correct environment external factors such as the correct amount of space for the crib, access to the appropriate power supply and a stable Wi-Fi connection, the user will be able to use the crib within a chosen room.

Ease-of-use is a top priority considering Tenshi's target users. However, in order to meet the user's need for a system that will remain reliable through the entirety of a child's infancy, Tenshi has made a compromise between out-of-box use and durability. As a result, users will need to be instructed on how to safely plug/connect system peripherals for cleaning purposes. After setup or cleaning, users will not need to interact with system sensor components and there will be no need for individual subsystem calibrations.

Users will be able to access data from the crib system through the Internet of Things (IoT) system device where only important analyzed data will be displayed. As a result, the system is restricted to users with compatible mobile devices that are connected to a stable Wi-FI connection. The mobile notification system interface will be intuitive and similar to other data analyzing applications (i.e. health applications) that are common on many mobile phones.

A.3. Technical Analysis

To ensure the system is designed for effective human utilization, it is important to take into account seven elements of UI interaction from Don Norman's "The Design of Everyday Things". This includes discoverability, feedback, conceptual models, affordances, signifiers, mappings, and constraints, each representing a different stage in design, outlining human factors [21].

A.3.1. Discoverability

Discoverability relates to the ease of user interaction from the current state of the device [21]. To extend, it is the degree of ease which the user can find system functionalities and features [22]. Therefore, the system interface will be limited to exactly what is required for the user to use in order to interact with the system efficiently yet effectively.

The Tenshi Baby Crib requires necessary steps of set up in order to effectively use all the functionalities that the system contains. A short list of setup instructions will be included with the Tenshi Baby Crib for the user to follow.

1. **Power on the Tenshi Baby Crib:** The Tenshi Baby Crib runs on power, so there will be a distinct switch to power on the crib.
2. **Download the Tenshi Baby Crib App:** The instruction manual will include exactly what app to download then the app will display the following steps.

The software setup system will be straight forward, as the app will take the user through a series of steps that they can follow to ensure that their corresponding crib works with their downloaded app. The user will be prompted to make an account then connect their account to their corresponding crib through instructions given on the app. Once it is connected, the user will be taken to the intuitive home page. The app will only start monitoring once a baby is placed in the crib. To support ease of discoverability, the user will be able to access all the main functionalities of the app from any view. This is supported through a stationed navigation bar at the bottom of the app, giving the user access to real time data on the home page, analytics of crib sessions over time, and settings view or adjustments.

A.3.2. Feedback

Feedback ensures that the system's state is comprehended by the user. Feedback is used as a means of communication to confirm that the system is functioning as assumed by the user as well as a notification to the user whenever specific states of the system changes design [21].

In regards to the crib, there will be feedback to the user to ensure that the crib is powered on through an LED light source. In short, if the crib is on, the light source will also be on, if it is off the light will be off.

As a monitoring system, feedback to the user is the focus of the app's functionality. Notifications will be sent depending on vulnerable thresholds set for baby position, room temperature readings, and noise levels. If the comfort range is exceeded for temperature and noise or position changes, notifications will be triggered. The following are examples of triggers for notifications:

1. Room temperature is too high.
2. Room temperature is too low.
3. Room is too loud.
4. Baby is awake.
5. Baby turned on their front.
6. Baby turned on their back.

The main source of user feedback will be completed through visualizations within the app. Real time data will be displayed on the app, through numbers, colours, and images of the system's state regarding baby position, noise level, or room temperature. If any vulnerable thresholds are reached by exceeding comfort thresholds, not only will a notification be sent to the user but the warning will also be visualized on the app. Both scenarios are highlighted in Figure A.3.2.1.

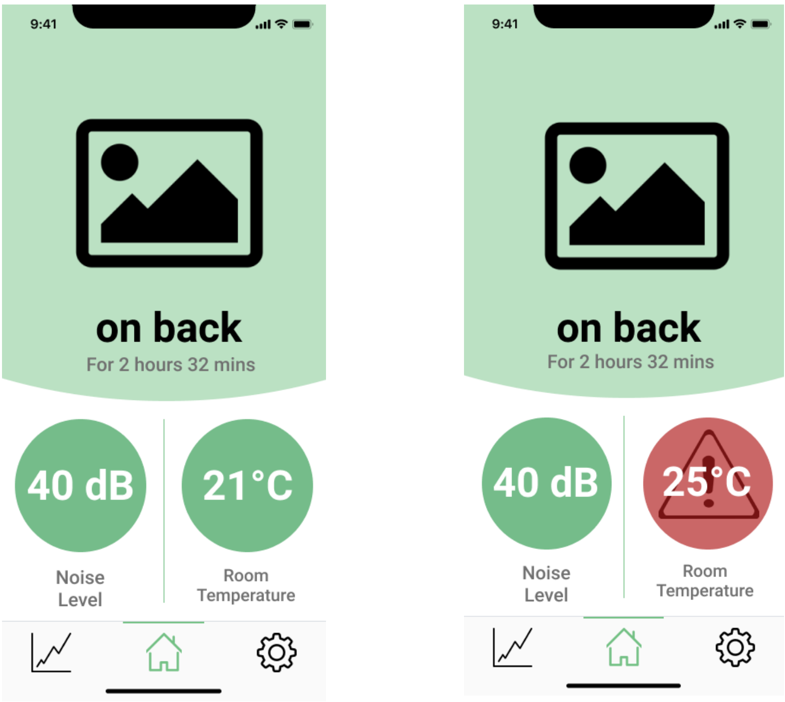

Figure A.3.2.1: Temperature, sound and positional feedback

A.3.3. Conceptual Models

The overall goal is to ensure that the designer conceptual model of the system is conveyed accurately for the user to form a model that is close to reality [21]. To define the designer conceptual model, the system consists of two main parts, this being the hardware system, Tenshi Baby Crib, and the software system which is the phone application. The sensor system is a subsystem of the Tenshi Baby Crib, as it will be integrated into the crib. Upon first setup of the crib, the user is made aware of the microcontroller along with the accompanying sensors. In pairing with the app, the user comes to understand the relationship between the sensor system and the phone software. To ensure that the designer and user conceptual models are equivalent, the Tenshi Baby Crib will present the conceptual model through the visuals of the phone app. The home page will make it obvious as to what values are being monitored for the baby, depicting its three sensor systems; temperature, noise, and position shown below in Figure A.3.3.1.

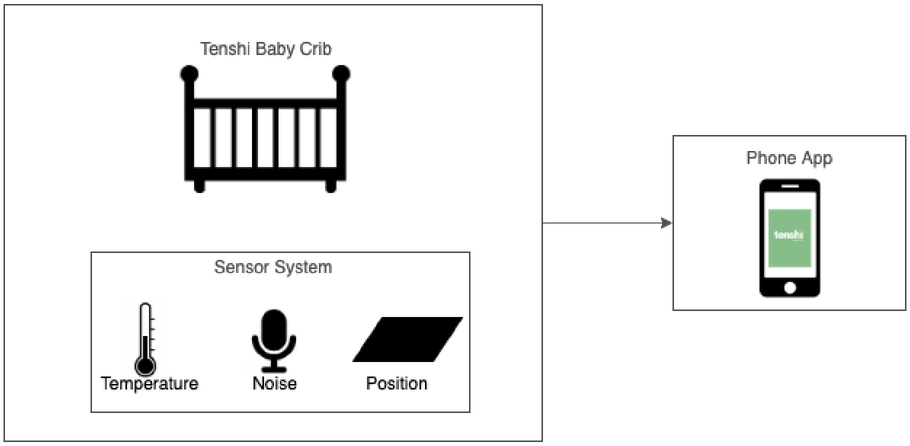

Figure A.3.3.1: System Diagram of Tenshi Baby Crib

A.3.4. Affordances

Affordances relate to the freedoms of the user to manipulate the interface by defining what actions are possible [21]. By limiting the user to make only the most pertinent, obvious and simple choices, the user experience can be streamlined.

Within the UI, it is done by focusing on informational views, where the user input is limited to navigation. To ensure intuitive navigation, the user is afforded with three main menu buttons on the bottom of the screen. These include the main menu, historical data, and settings depicted in Figure A.3.4.1. By persisting in every view, the user can swap between menus seamlessly.

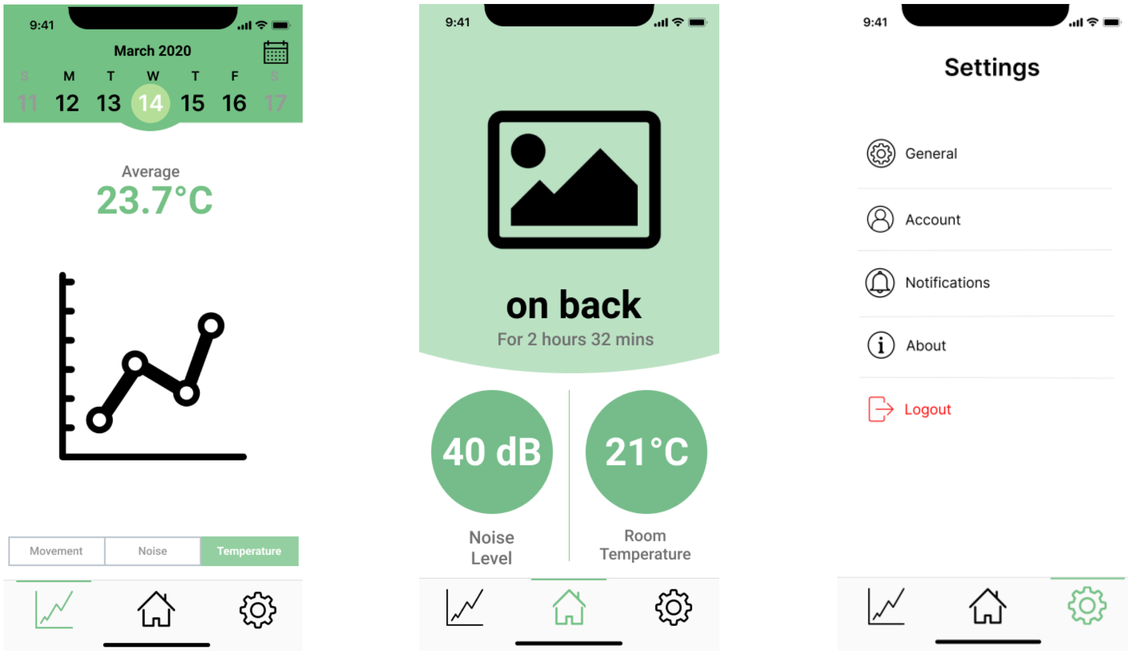

Figure A.3.4.1: Main User Menus

Within the historical data menu, users are afforded the option to change the date to show the corresponding data. The menu that requires user input is the settings menu. There, users will be able to turn off and on functionality such as notifications, and alter the thresholds for a given monitoring indicator.

In operating the crib hardware, the user will incur no overhead in initiating monitoring. This is because monitoring automatically commences once the baby is placed in the crib, as indicated in the UI. In this way, the affordances in using the product are aligned with a standard crib. Lasty, the user is afforded the option to detach the pressure pad for cleaning.

A.3.5. Signifiers

Signifiers bridge the gap between actual affordances and perceived affordances [21]. By providing visual cues as to what the user may do, less is left up to users' interpretation. Within the UI, buttons are presented as icons or with labeled text. Lines and highlighting are utilized to indicate the currently selected option and divide sections as shown in Figure A.3.5.1.

Figure A.3.5.1: Menu Buttons Icons with Highlighted Selection Signifiers

On the settings view, green on the switch indicates a feature is enabled and can be disabled, and a slider is used to indicate pickable threshold values.

When it comes to detaching the pressure-pad for cleaning, arrows showing where the pad can be attached or detached, and the markings showing how to do so, will be placed accordingly.

A.3.6. Mappings

The way elements are mapped gives insight as to how the user can interact with them [21]. By using intuitive elements, information can be provided in a non-intrusive way. In the UI, it is done so in a number of ways. Icons on buttons describe their corresponding menu: home icon for the main screen, chart for historical data, cogwheel for settings. In the historical data view, a button is provided which creates a calendar pop-up upon pressing. By doing so, users understand immediately that choosing a date will show the corresponding data. In the settings view, simple toggle switches are used to enable or disable a function, and sliders are used to offer potential values for thresholds. The settings view is depicted below in Figure A.3.6.1.

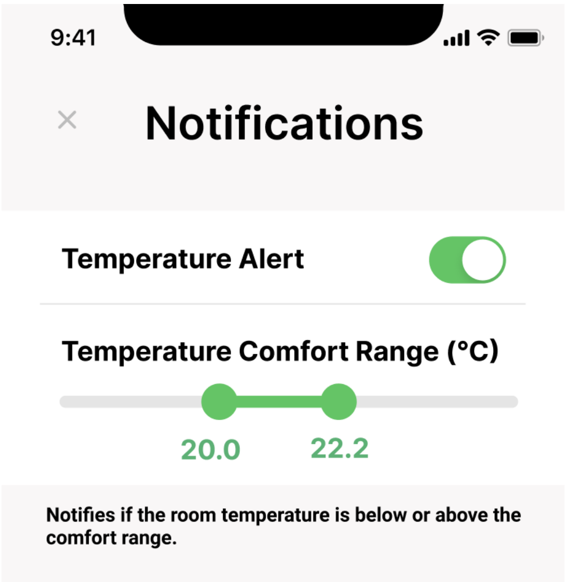

Figure A.3.6.1: Notification Settings with Switch Toggle and Slider Elements

A.3.7. Constraints

Constraints are necessary to enforce that the system is being used as intended, limiting errors [21]. Hence, it is important to constrain user choices to only those that are required.

There are physical constraints to the baby crib such as an integrated sensor system. The noise detection and temperature detection will be directly installed into the crib such that user tampering is deterred. The cribs area constraint for the baby is designed with safety and comfort in mind.

Constraints to the app include elements with limited amount of choices. Use of user-input values are avoided, as pre-set values for users to choose from are provided. Switches and buttons give linear choices and in no way interact with each other.

A.4. Engineering Standards

User Interface and Appearance Design development will adhere to the following engineering standards and mobile platform standards throughout the project timeline. This will ensure that visual, audio, and touch based interactive elements of the product will be tested to be safe, consistent and reliable. Below in Table A.4.1 and A.4.2 are the standards and guidelines that will be followed.

Table A.4.1: User Interface and Appearance Design Standards

Standard	Title
ISO/IEC/IEEE 12207-2017	Systems and Software Engineering - Software Life Cycle Processes [23]
ISO/IEC 11581-1:2011	Information Technology - User Interface Icons - Part 1: Introduction to and Overview of Icon Standards [24]
IEC 61508-1:2010	Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems - Part 1: General Requirements [25]
ISO 9241-161:2016	Ergonomics of Human-System Interaction - Part 161: Guidance on Visual User-Interface Elements [26]
ISO 7779:2018	Acoustics - Measurement of Airborne Noise Emitted By Information Technology and Telecommunications Equipment [27]
UL 2595	General Requirements for Battery Powered Appliances [28]
SOR/2016-152	Cribs, Cradles, and Bassinets Regulations [29]
IEC 60950-1:2005	Information Technology Equipment - Safety-Part 1: General Requirements [30]
IEC 62047-33:2019	Semiconductor Devices - Micro-electromechanical Devices - Part 33: MEMS Piezoresistive Pressure-Sensitive Device [31]

Table A.4.2: Mobile Application Guidelines

Provider	Guideline

Apple	Human Interface Guidelines [32]
Android	Core App Quality [33]

A.5. Analytical Usability Testing

The Tenshi company will perform heuristic evaluations to verify product usability [34]. In depth testing must be conducted by system designers first in order to get relevant feedback from the targeted users' usability tests. Conducting analytical usability testing and reviewing the feedback is done to address the critical issues in the UI usability given that system developers have a comprehensive understanding of the system's functionality. The goal of performing analytical tests is to identify neglected use cases, bugs and inconsistencies that would render the user's experience of the system to be nonviable.

The heuristics that will be evaluated will be the learnability, efficiency, memorability, error, and satisfaction of each interactive element of the system [35]. These heuristics will be standardized to a numerical scale from one to five where one is a major problem and five a non-issue. These usability tests will enable optimization of the following interactive elements: setting up the crib, detaching the pressure mat, creating an account, logging in, browsing through the application, and viewing data.

Developers will conduct these heuristic evaluations on each subsystem separately then on the system as a whole. It is practical and cost efficient for developers to conduct these white-box tests prior to external, targeted users as there is less time spent explaining the system and less time spent testing expected use cases. As a result, intentional testing can be done for additional edge cases and as well as aspects of the experience that would be out of the scope of the target audience. These aspects would include safety and reliability. Analytical usability testing also presents an opportunity for developers to see their designed component interact with the entire system. Lastly, developers with context to the system are able to provide possible solutions to the problems they encounter.

Tenshi will prioritize utilizing the feedback from these analytical usability tests to make improvements to the UI before allowing external testing to occur.

A.6. Empirical Usability Testing

Performing empirical usability testing will involve potential users and those that have already raised newborn children. These tests will revolve around observing user interactions in regards to the crib and the mobile application with a brief explanation of the product. During the tests, designers will log user actions, measure a number of metrics, and inquire for feedback. Thus, analyzing user interactions alongside quantifiable measures will enable designers to determine if the product clearly achieves design goals and validates the analytical usability testing results. Table A.6.1 below outlines the sequence of actions to be analyzed that the designers will instruct the users.

Table A.6.1: Empirical Usability Testing Guideline

Designer Prompt	User Action	Measures
Designer will introduce the "Tenshi Baby Crib" as a baby monitor. Designer will give the user the instruction manual for setup.		
Connect the pressure mat	User grabs the cable and connects the microcontroller to the pressure mat	Time: User connects pressure mat. Error: User finds it difficult to connect.
Turn on the crib	User flicks the power switch.	Time: User finds power switch. Error: User questions if everything is connected. Error: User questions if the crib is on.
Connect the crib to the app	User connects the crib to the app by following set up instructions on the app.	Time: User connects crib to the app. Error: Crib cannot be connected to the app.
Designer will place a baby doll in the crib. Designer will provide the user with a smartphone and prepared login information.		
Open up the application and log in	User types in the username and password.	Time: User successfully logs in Error: Incorrect username or password.
Explore basic app functionalities.	User finds historical data, current information, and settings views.	Time: User successfully understands functionality of application. Error: User questions functionalities of the application.
Change all adjustable app settings	User adjusts notifications and threshold values.	Time: User updates and changes all settings. Error: User is unable to figure out how to adjust settings.
Designer will give the user clean up instructions.		
Detach the pressure mat and prepare it for washing.	User follows instructions and removes the washable pad.	Time: User removes washable material of the pad. Error: User finds it difficult to remove the material.
Designer will inquire for feedback from the user for each step.		

By the end of performing these tests outlined in Table A.6.1, designers will have gathered the amount of time to perform all actions, the number of errors that occur, methods of error recovery, and general user feedback. This information will allow further optimizations to the user interface and appearance design of the Tenshi Baby Crib.

A.7. Conclusion

The interactive elements of the Tenshi Baby Crib are integral to product success. Following the analysis and specifications of this appendix ensures optimizations will be continuously made for user usability. Given that the target users—parents and caretakers—are expected to utilize the system through the entirety of a child's infancy, durability and simplicity in interactions are prioritized in order to seamlessly incorporate the system into their daily routine. To achieve this, crib setup and the mobile application user interface have been designed to clearly represent the purpose of the system and allow users to be in full control of the application with minimal errors. These appearance design decisions will be tested against a multitude of standards and guidelines, and undergo extensive usability testing to verify the system's intended interaction behaviours. As a result, the analysis and design decisions outlined in this document will facilitate users to utilize all features offered by the Tenshi Baby Crib.

Appendix B: Test Plan

B.1. Scope

This appendix outlines the testing strategy employed during development of the Tenshi Baby Crib in the proof of concept and prototype phases. It covers the required items needed to accurately verify the functionality of the product and facilitates accountability. Performing the testing strategy alongside a uniform test plan will enable designers to pinpoint issues and ensure verification of all system features.

B.2. Testing Strategy

Meeting the requirements set out by the Requirements Specification is paramount to maximizing product quality. As such, the test plan shown in section B.4 lists the test environment details, requirements and their tests to verify they have been met.

To facilitate efficient issue tracking, the test engineer must detail the date the tests were performed, their name, and the type of test being performed. The type of test selected dictates which tests are performed. Combining these details enables designers to verify that prior issues have been resolved, and for any new issues that occur to be accurately tracked. Moreover, the test plan is used to sign off on project milestones for certain features, proof of concept, or prototype phases. Once a test engineer has completed a test plan, the results must be discussed and approved by a designer to ensure the tests have been performed exhaustively so as to find the root cause of issues.

This method of uniform testing facilitates efficient, accurate, and exhaustive test results to drive development forward.

B.3. Tools Required

The tests outlined in section B.4 requires tools to verify design circuitry and usability against requirements. The required tools for each section and their purpose are listed below in Table B.3.1.

Table B.3.1: Required Items for Performing Tests

Test Section	Item Required	Purpose
Temperature Monitoring	Digital Thermometer	Accurate comparison for temperature measurements.
Noise Measuring	Sound Pressure Level Meter	Calibrate microphone and perform comparison for noise measurements.
Position Detection	Test Weights, Baby Doll	Reliable results and analysis for weight and position detection.
All	Mobile Phone or Computer	Required to interface with design circuitry and test mobile application.

B.4. Tenshi Baby Crib Test Plan

Table B.4.1: Tenshi Baby Crib Test Plan

Tenshi Baby Crib Test Plan			
Date Performed:		**Test Type (circle):**	Unit Test System Integration Test Proof of Concept Test Prototype Test Regression Test
Performed By:			
Approval Signature:			
Requirement	**Test**	**Result**	**Comments**
Temperature Monitoring			
R3.2.1-POC-1 R3.2.2-POC-1 R3.2.2-POC-2 R3.2.2-POC-3 R3.3.1.2-POC-2	Thermistor reading is accurate to ±1°C		
	Thermistor reads ≤ 16°C		
	Thermistor reads ≥ 23°C		
	Microcontroller reads 300 temperature measurements in 5 minutes.		
R3.2.2-PT-9 R3.3.1.1-PT-2	Software detects vulnerability of ≤ 20°C and ≥ 23°C within 5 seconds.		
Noise Measuring			
R3.2.1-POC-2 R3.2.2-POC-1 R3.2.2-POC-2 R3.2.2-POC-4 R3.3.1.2-POC-2	Noise reading is accurate to ±1dB		
	Noise reads ≤ 40dB		
	Noise reads ≥ 120dB		
	Microcontroller reads 300 noise measurements in 5 minutes.		
R3.2.2-PT-10 R3.3.1.1-PT-3	Software detects vulnerability of ≥ 75 dB within 5 seconds.		
Position Detection			
R3.2.1-POC-3	Weight reading is accurate within ±10%		

R3.2.1-POC-4 R3.2.2-POC-1 R3.2.2-POC-2 R3.3.1.2-POC-2	Pressure contact can sample unique readings in the range of [0kg, 1kg] of applied pressure with a 3kg mass object		
	Microcontroller reads 300 pressure samples in 5 minutes.		
R3.2.2-PT-6 R3.2.2-PT-7 R3.2.2-PT-11 R3.3.1.1-PT-1	Software detects baby doll is on their back		
	Software detects baby doll is on their front		
	Software detects baby doll rolls over within 5 seconds		
Crib to Mobile Application Communication			
R3.2.2-PT-12	Mobile application receives notification from the Tenshi Baby Crib whilst on the same network.		
R3.3.1.2-PT-3	Mobile application indicates that there is no connection to the crib whilst the crib is off		
R3.3.1.2-PT-4	Mobile application indicates that the pressure mat is disconnected		
R3.3.1.3-PT-1 R3.3.1.3-PT-2	Data from the crib is only read and updated whilst the corresponding account is logged into the mobile application		
R3.3.1.2-POC-1	**Above PoC tests passed in a home environment of 16°C to 22°C**		
Product Usability			
R3.2.2-POC-5	Mobile application displays current temperature and temperature over time graph		
	Mobile application displays current noise level and noise over time graph		
	Mobile application displays current baby position and baby position over time logs.		
R3.2.2-PT-8	Mobile application displays above datasets in hourly, daily, and weekly ranges.		

R3.3.1.4-PT-1	Pressure mat functional after connecting and disconnecting the cable to the microcontroller 100 times.		
R3.3.1.4-PT-2 R3.3.1.4-PT-3	Discoverability: In the mobile application, all views are accessible from any state via the navigation bar		
	Feedback: Flicking the power switch on turns on the crib LED.		
	Feedback: All elements shown in Figure A.3.2.1 are functional.		
	Conceptual Model: Instruction manual accurately depicts the system as shown in Figure A.3.3.1		
	Affordances: All elements shown in Figure A.3.4.1 are displayed.		
	Signifiers: Selected options are highlighted as shown in Figure A.3.5.1		
	Mappings: Sliders and Toggles in Settings are visually functional.		
	Constraints: Noise and Temperature circuitry are not visible.		
R4.1-PT-1	Instruction manual accurately describes all steps to dismantle crib electronics.		
R4.1-PT-2	Microcontroller does not sample data from the thermistor when temperature notifications are off.		
	Microcontroller does not sample noise readings when noise notifications are off.		

	Microcontroller does not sample pressure readings when pressure notifications are off.		
Additional Comments			

Appendix C: Design Options

C.1. Temperature Options

Several temperature sensor options were researched and compared, as reliable temperature readings are dependent on the sensor choice that suits the application for the Tenshi Baby Crib. Before comparison, the requirements to suit the Tenshi system regarding the temperatures operating range, accuracy, stability, and sensitivity along with response time were decided. The four sensor options considered were thermocouples, resistance temperature detector (RTD), temperature IC, and the thermistor. The thermistor was chosen as it fulfilled all the requirements for the project.

Through comparison, all the advantages and disadvantages of each temperature sensor were considered to make the best decision. Thermocouples are great for wide temperature ranges but do not provide great accuracy. RTDs are very accurate and stable however provide the slowest response times. Temperature ICs are easy to use as they output digital values however they are fragile and limited in speed as well.

The chosen thermistor is a 10k NTC thermistor with a B constant of 3950 ±1%. Although it has a limited operating range, a wide operating range is not necessary for the Tenshi Baby Crib as the monitoring system focuses on house ambient temperatures, ranging from less than 16°C to greater than 23°C, which the thermistor accurately covers specifically to within 1% accuracy at 25°C. Additionally, the thermistor is highly sensitive along with fast response times, evident since thermistors are frequently used in appliances such as fire alarms. It is also greatly resistant to shock and the chosen thermistor is waterproof, ensuring its long term stability.

C.2. Sound Options

Prior to designing and creating the aforementioned sound sensor circuit; various sensor modules and components were considered. Alongside the module's performance and specification, the chosen microcontroller's ADC was considered.

A shortlist of sound sensors were selected for their compatibility with Arduino and similar MCUs. The first sound sensor was the *Velleman* sound sensor module with on-board potentiometer and LM386 op-amp. The second sound sensor considered was the *OSEPP* sound sensor which utilizes an LM286 op-amp as well. Both chips use a 50Hz to 20 kHz electret microphone.

The first design disadvantage of using these pre-built sensors include the inability to adjust the gain of the circuit. The *Velleman*'s potentiometer was difficult to handle and move precisely to adjust the gain. Therefore, the detectable sound pressure range is limited with these sensors.

The second design disadvantage includes the inability to add specific high- and low-pass filters to the circuits. Conducting testing on the aforementioned sensors yielded the expected performance of a basic clap sensor with a short listening range. Sampling with these sensors and averaging the data resulted in a relatively noisy signal that would not suffice for the critical component within the sound detection subsystem.

Considering and analyzing these circuits allowed Tenshi to establish a baseline of performance for the sound detector subsystem.

C.3. Position Detection Options

While researching ways to detect sleeping position, several options were weighed before the use of a pressure mat was decided. A depth camera to detect position was the main consideration. Using a camera to detect position depth images would need to be analyzed with a machine learning algorithm. This introduces several layers of complexity that requires time and expertise. Also, to ensure accurate and reliable results, the camera must be set up in a precise manner. Incorrect set-up may introduce false positives and detection of movement not related to sleeping position. Thus, relying on the user to set up the camera in an optimal location not only constrains how the user can use the Tenshi Baby Crib, but also introduces variability in results. Lastly, our goal is to create a product with an experience and look as close to a standard crib as possible. Mounting a camera may not be feasible for all users, and certainly detracts from the overall feel and comfort of the room.

By using a pressure mat, the step of filtering erroneous data is avoided. Focusing on the degree of pressure and its location, data directly related to position is provided. This means that the 2D array of pressure-levels produced can be input to machine learning algorithms, outlined previously in section 4.4. Additionally, no extra setup needs to be done other than placing the mat in the crib, and plugging it into the microcontroller. In this way, operation of the Tenshi Baby Crib does not deviate too far from standard cribs. Lastly, the sanctity and comfort of the room is preserved with no intrusive technology within view.

C.4. Microcontroller Options

There were four main microcontrollers that were researched when searching for a microcontroller. Those four were the Arduino Uno, Arduino Uno Wifi Rev2, Arduino Due, and the ESP32. All the microcontrollers were assessed in four sections which are seen below in Table C.4.1.

Table C.4.1: Microcontroller Comparisons

	Connects with wifi	Computational power	Lots of IO pins	Sample multiple sensors
Arduino Uno	Needs Module	Decent	No	Yes
Arduino Uno Wifi Rev2	Yes	Decent	No	Yes
Arduino Due	Needs Module	Strongest	Yes	Yes
ESP32	Yes	Strong	Yes	Yes

As seen in the results, the strongest debate was between the Arduino Due and the ESP32. The question is, which is more important, computing power or better integration to wifi. The Tenshi team agreed to choose the ESP32 over the Arduino Due because of the built in wifi module.

By having the Wi-Fi module built in, no extra expenses were needed for an external wifi module. Furthermore, strong computing power is important; however, the amount of computing power the ESP32 presents is already more than enough for the Tenshi Baby Crib. Using the Arduino Due would be unnecessary. Lastly, the ESP32 has a dual core microprocessor which beats out all the Arduino microcontrollers in terms of multiprocessing which would be greatly needed for all the sensor sampling required.

C.5. Notification System Options

C.5.1. Framework Options

After considering framework options for the mobile app, cross platform development was found to be ideal. This requirement allows the Tenshi Baby Crib to be accessible for both android and apple users. If the framework could not support cross platform development, the target market would be biased based on what the framework supports. Furthermore, by having cross platform development it would reduce the codebase to one which would result with easier maintenance and support.

The two cross platform frameworks that were considered were Flutter which was made by Google and React Native which was made by Facebook. One of the key factors for deciding was which framework has better documentation and a bigger development community to support resources. Since React Native was released earlier than Flutter, it has more libraries, documentation, and stable versions [36]. Additionally, even though the Tenshi team does not have experience with either framework, the team is experienced with React.js which is very similar to React Native. This also applies with the languages each framework uses. React Native uses JavaScript and Flutter uses Dart. Of these two coding languages, JavaScript was more well known to the team. Reflecting on the needs of the Tenshi Baby Crib and the learning curve of each framework, React Native was the clear choice for the cross platform framework.

C.5.2. API Options

Creating an API is new for the Tenshi team. When researching different ways of creating an API, the most popular option was using the Amazon API Gateway (AAG). Another option was to create an API using python. The biggest factor when considering options that the Tenshi team has no experience or preference with is how good the documentation and developer community is.

Although the team does not have experience with either option, the team does have some experience with Amazon Web Services (AWS). This gave the AAG option the slight edge over creating the API with python. However, when researching online examples and documentation about AAG, there were far more examples and online support than there was for python. The creation of an API using python is possible however there are multiple ways to do so using other frameworks such as flask. Learning another framework was extra work which could be avoided if the team went with the AAG implementation. Ultimately, when reviewing the pros and cons of both options, the Amazon API Gateway was the better choice for creating the API.

C.5.3. Database Options

For the Tenshi Baby Crib, a database is essential for the notification system. The functionality of the database however is rather simple and straightforward. The database would store and retrieve data without any special concerns for unique data types. Any database could work well with the Tenshi Baby Crib so it came down to preference. The Tenshi team has experience with databases, more specifically relational databases.

The two databases that were closely debated on utilizing was mySQL and PostgreSQL, both of which are relational databases. The deciding factor between these two databases was the fact that PostgreSQL has a native JSON parser. With how frequent the sensor system will be sampling, a large volume of data will be stored and retrieved. JSON works side by side with JavaScript and having a native JSON parser will help development become more efficient and robust. Hence the reason PostgreSQL was chosen as the Tenshi Baby Crib database.

C.5.4. Graph Library Options

With React Native being a new framework for the team, the graph library option was very difficult to gauge. Once again, the biggest deciding factor was based on the documentation and developer community for a particular graph library. There were three graph libraries that stood out and were compared. Those three were Victory Native, React Native Chart Kit, and React Native SVG Charts. All libraries have the potential of creating aesthetic graphs that would work for the mobile app. Nonetheless, Victory Native was the most supported library with multiple online examples, easily understandable documentation, and overall popularity. Those three reasons put Victory Native slightly over the other two graph libraries and the reason why it was chosen for graph visualization.