# COEN 4920- Principles of Design 2024

Senior Design Final Report

Due: Friday, May 3rd, 2024

Project (#C41) – System for Detection of Forgotten Children in Cars

Viviana Garcia, Ben Minick, Danny Okerlund, Julia Roesler, Mitchell Rouse

Faculty Advisor: Dr. Cristinel Ababei

# Table Of Contents

# Executive Summary

## Overview

Our team was tasked this year with creating a child safety device that will determine if a child is left behind in a vehicle when a parent exits. There are other devices that are available on the market, so we wanted to make our device as versatile and compatible with as many different cars as possible.

## Problem Summary

Each year, children die or become endangered because they are left unattended in their vehicle by their parent or guardian. Cars in direct sunlight can rapidly increase in temperature, reaching unsafe levels in just over 2 minutes. The purpose of this device is to create a fast and reliable way to illicit a response from the parent or guardian to return to their vehicle and child.

## Solution Summary

Our final prototype is an additional car seat mat connected to an electronic housing that, using weight and temperature sensors, can accurately notify a previously Bluetooth connected device that there is a child still sitting in the car seat. The electronic housing contains an HC-05 Bluetooth module and an Arduino Nano BLE that are in constant communication with a connected Android cell phone. When the cell phone reaches a distance away where the Bluetooth module is no longer able to keep a connection, the designed AutoAngel app will know if there still was a child detected when the cell phone disconnected. If a child was detected, an alert will immediately be sent to the cell phone to notify the owner to return to their vehicle. If there was not a child detected, no alert will be sent to the cell phone and the owner can proceed with their tasks as normal.

Testing for this device showed that a user was able to turn off their vehicle, exit, and receive a notification that a child was still detected in the vehicle less than 2 minutes from turning off the car to walking a distance where the Bluetooth device disconnects. This creates optimal safety for the child and gives the parent or guardian peace of mind where they are able to have a backup reminder for additional child safety.

## Financial Summary

At the beginning of this project, we were given a total budget of five hundred dollars. After all materials bought throughout the semester for different variations of the design, we only spent $366.89. This total made us underbudget and successful as we stayed within the financial constraints of the project. The total cost for the material of the final prototype was $234.17 and works as expected.

## Discussion of Legal Issues

The biggest legal concern that we had to consider were the legal implications of infringing on intellectual property. To begin the project, concept searches were conducted in search of prior art and products currently available on the market. The team dove deep into research to investigate what has been done before and what is out there. Current systems include alarms that are based on a variety of sensors implemented into car seats, external devices that can be attached to car seats, and systems that are implemented to change the state of the car to make it safer for the child left behind. For example, U.S. patents US7714737B1, US9139128B1, and US9569948B1 describe car seats and car seat additions that are designed to communicate with the vehicle's alarm system when it is determined that a child has been left in a car. Another example can be seen in US10556581B2 where the existing car electronics are used in conjunction with other sensors to automatically trigger the vehicles air conditioning if the conditions inside become unsafe. Some sensors used include motion detection, CO2 detection devices, and pressure sensors. Our focus was to design a system that uses a multitude of sensors to provide notifications so that the driver and others are aware of the dangerous situation.

There are a few products currently on the market that try to attack the problem at hand.

1) Patent US9569948B – thermal and weight sensors, sets off car horn and flashes car lights. Not wireless, complex and messy to install into vehicle.

2) Patent US9139128B1 – weight sensor in car seat, sets off car horn, only checks weight, no other factors.

3) Patent US77714737B1 – weight and seatbelt sensor, sets off car alarm, attaches to OEM car parts so it may be difficult to connect and setup.

4) Patent US10556581B2 – software detection system using CO2 levels, infrared energy, and temperature, sets off car alarm, no hardware – would come built into vehicle.

5) Amazon Product 'Ride&Remind' – backdoor sensor, sets off an alarm inside the vehicle, has a lot of false positives

6) Amazon Product 'Little Helper on Board' – weight sensor, set off light and sound from the small device attached to the smoker outlet, extremely simple way to solve the problem at hand.

Our goal was to make the system from low-cost hardware and focus on communications with a driver's smartphone, not the vehicle. After looking at this prior art we were ambitious to begin designing our system. The goal was to pick multiple ways of detecting a child to ensure less false positives through multiple sensor checks. Doing this while also implementing wireless communication for parents to get informed from their own cell phones about the child left behind is something that this market needs. Current systems that are available offer similar features but do not include features of being based on Bluetooth smart phone proximity to the car and a variety of in seat sensors.

In the state of Wisconsin, there are no specific laws that dictate the penalty for a parent or guardian for leaving their kid in a car. However, there are currently 21 states in the United States that explicitly say that it is illegal to leave a child unattended in a vehicle. Specifications vary by state. There is also no federal law that dictates a penalty. If a child dies because of being left unattended in a hot car, it is considered a class G felony which is punishable with up to 10 years in prison or a maximum of a $25,000 fine [3].

# Design Justification

In the final design, the necessities that were being assessed were the real-time monitoring to know if a child is in the car, a dual-factor alert system with temperature and weight sensing, and a user-friendly and easy-to-use interface. It was crucial to provide the customer with the most optimal and effortless interface to ensure it can be used even in a high-stress situation. It was also important that our product could be compatible with multiple different types of vehicles and was easy to move between vehicles with a car seat.

Anyone who has a young child and a vehicle would benefit from the use of our product, but we do not have a direct customer other than our advisor Dr. Cristinel Ababei. To ensure the happiness of our customers the integration of our system will not interfere with any typical use of the vehicle such as turning off/on the vehicle, having groceries or luggage on the product, or just temporarily stepping out of the vehicle. The customer needed to know they can use this product with no change in daily life while ensuring the safety of their children. When designing the product, we needed to address multiple concerns to prevent interfering with any customer needs.

The first of these concerns was how we were going to come up with a design that did not infringe on any current intellectual property. The first three weeks of the semester were focused mainly on researching prior art and current systems on the market. Our conducted search results can be seen in the discussion of legal issues where we go into detail over what we found and the sensors and implementation. Based on what we found for our prior art research, we then knew what design concepts were and were not novel.

The next concern that we needed to address was deciding what sensors we were going to use for our final design implementation. In terms of choosing which sensors, we were going to implement, the options we had to consider were weight sensors, strain gauge, temperature sensors, CO2 sensors, infrared sensor, camera, and microphone.

Weight sensors were everyone's first consideration when proposing a design. It was the simplest sensor that could detect a child sitting within a car seat.  If there is a child on the weight sensor, then there is a change in an analog signal to the connected microcontroller. If there is not a child, then no change in the signal will be detected. There are false alarms that need to be considered when using weight sensors as there is no way for them to detect by themselves if it is being triggered by a human or an object. Therefore, only relying on weight sensors would not be a practical design decision. For our final design implementation, we used strip resistive pressure sensors to act as one of two main sensors.

Temperature sensors were our second consideration when thinking about the best design implementation. A typical human body temperature ranges from 97 to 99 degrees Fahrenheit. A temperature sensor would be able to determine if there was a child or human present based on a change in seat temperature being warmed from the human body. The false alarms that we would have to consider with this type of sensor would be the temperature of a hot car in a hot climate triggering the temperature sensor. This was a major consideration on our mind when we were drawing up our concepts for our design as we wanted to have the most reliable design possible. Ultimately, we decided to continue using the temperature sensor because of its advantage of being able to detect and differentiate a human from another object being left in the car. We believe that the advantages of using a temperature sensor to properly detect a child outweighed the disadvantages of possible false positives being sent to a person's smartphone.

Overall, we decided to set up our Arduino code where the condition of whether a child is left in a car or not is triggered by the combination of inputs from both the weight and temperature sensor. If both the weight and temperature sensor are considered triggered based on thresholds defined in the code, then it is determined that a child is detected. The combination of the sensors will limit the disadvantages of both the sensors. The temperature sensor is able to solve the weight sensors disadvantage of detecting the body temperature and determine a human versus and object and the weight sensor would be able to check the temperature sensor if the sensor was triggered by a hot car but there was no child in the car seat.

In our design, weight and temperature sensors were not the only type of sensors that we considered. We also debated the use of a CO2 sensor, an infrared sensor, and the use of a camera or microphone.

CO2 sensors would allow us to determine if there was a human actively decreasing the oxygen levels in a car. If there was an increase in CO2, then the connected microcontroller would be notified that there is a child in a car. From our research we determined that the use of said sensor would not be reliable enough and would be a steep investment. The use of CO2 sensors would not be quick enough to trigger and alert a parent that a child has been left behind in a car. Over time as a child sits in the car, the CO2 levels would increase, but the change in CO2 would be too gradual to trigger an alert to a parent in an appropriate amount of time unless an extremely specified sensor was used. We needed to use a sensor that could give us a definitive, rapid result in the shortest amount of time possible. The placement of a CO2 sensor also caused a design debate because it would have to be placed close to a child's head and face to get an accurate rating. Overall, the disadvantages of using a CO2 sensor outweighed the advantages.

We debated the use of an infrared sensor or thermometer for a while before we opted not to use it. Our goal for this project was to have the simplest implementation and installation for a user as possible. The use of an infrared sensor would require mounting on the back of the front seat of the car by the user to reliably face the car seat. This would raise problems of wiring, communication, aesthetics, and safety for a child. We wanted to design a device with the least number of lose wires possible as these pose a risk for any small children. Electrically, the use of an infrared sensor would be beneficial and allow for our device to be more accurate, however mechanically, it poses more safety issues and risks for the child that we did not want to infringe upon. It also requires the user to properly set up the sensor to accurately point toward a child car seat and not incorrectly install the device. The risks with the set up and safety hazards made us ultimately decide that this was not the best sensor for our project.

We ran into similar issues as the infrared sensors when we were debating the use of a camera monitoring a child in a car seat. The use of a camera to see if there was a child left in a car would face the same wiring and set up issues as an infrared sensor and pose the same safety concerns. We also would have had to design a way to either stream a live feed from a camera to a connected cell phone or send updated pictures intermittently, which would increase cost and complexity to our system. The use of a video feed or pictures also brings up cybersecurity concerns regarding the child's safety. We did not want to design a system where there was potential of a hacker getting access to video or photo of a child in their car seat and take advantage of the fact that a child has been left in a car by themselves. The complexity this would add to the system and the additional safety concerns were not consistent with our project objective and this is why ultimately, we opted not to use a camera as part of our design.

Finally, the last sensor implementation that we considered was the use of a microphone. When a child is active and crying from being left in the car, this would be a good implementation to use as an active microphone could trigger a condition that the child was left in a car. However, if a child is sleeping or in a state of distress such as heat stroke, they would not be making noise or active enough to trigger that they have been left behind. The use of a microphone would be similar to the use of a CO2 sensor where it would be able to pick up on a change and alert the parent, however neither would be consistent enough to reliably let the parent know if the child has been left behind in a car.  Microphones also pose similar cybersecurity concerns as a camera. We do not want to include a live microphone feed to be transmitted to our app that could potentially be hacked or intercepted putting the child in more danger.

With the consideration of various number of sensors and also considering the factors of mechanical design, child safety, and odds of successful implementation, the combination of the weight and temperature sensor was the best in all categories, and this is what we used in our final design.

As we were coming up with our design, we considered the concept of using preexisting technologies already implemented in the car. We could have used similar technologies such as airbag detection triggers to figure out if there is a person or child currently in a seat. However, most cars do not have airbag detection in the backseats of their car and would require new technology to be implemented in the back seats. We also considered the idea of using our weight and temperature sensor to trigger the car's emergency alert alarm. This would enable the use of the car horn and lights and require attention by a parent or bystander to disable the alarm. The use of all of this technology is very specific to a car's make and model and would require us to talk with car manufacturers to implement our systems into their cars. It also would limit the use of our product in any older model of vehicle and would be constrained to new cars. Our top priority is child safety and even if there was a way for us to successfully find a way to sync our device with a car's panic alarm, we did not want to have a system that would notify the public that a child was alone in a car. We wanted to create a system that would only notify a parent or guardian. Our senior design team does not have the clearance nor the budget to implement a system like this. We wanted to create a versatile device that could be put into any car that was safe and practical for both a child and their parent or guardian.

When designing the hardware, we had to find a balance between the complexity and set up of a device with the accuracy of the device. It was not in our best interest to design an extremely complex system because of our lack of an industry sponsor and the time constraint of only 1 school year to complete this project. Designing a complex device would allow us to have higher accuracy in our detection. However, we stand behind our simpler design choices to make the product as low cost and accessible to as many customers as possible.

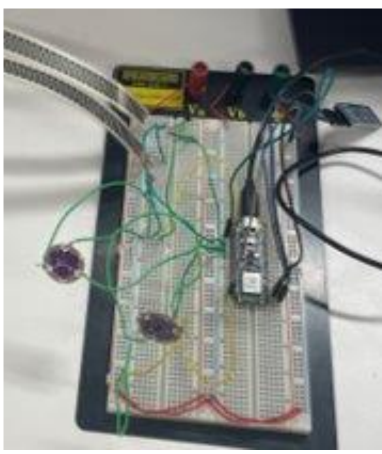
The final major design choice we had to make was how we were going to design and implement an app to properly notify the user of a child detected in a car. The first decision was whether we were going to create an Android or IOS app. After our research and talking with our advisor, we determined that at the time it was best to create the app on a free desktop software called Android studio where we can get familiar with the Android app development easily. There are countless examples online on how to implement this to create an Android app and as all of us as team members have windows-OS laptops, this seemed like the easiest development tool. Building an app on IOS would have required us to follow harder constraints and security through Apple and would make debugging more difficult.

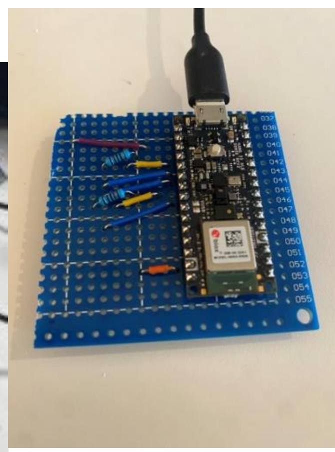
In the final design, the Bluetooth module is sending a status to the Android phone every 0.5 seconds as to whether the sensors are notifying that a child is detected in the car. When the smart phone disconnects from the Bluetooth module because of being out of range, the last known signal is read by the app to determine if a child was detected in the car. A 0 signal would represent that there is no child detected and the app that we design in Android Studio should not respond in any way. A 1 signal would indicate that there was a child left behind in the car and a push notification is sent to the parent or guardian's cell phone. This notification will indicate that the parent must return to the car to care for their child.

# Final Detailed Design

Hardware: Electrical Design

Our hardware design began with simple sensor integration onto a breadboard. We needed to confirm that our sensors were working properly and learn what values they were detecting. The Arduino studio interface included base code for both digital and analog sensors. Both our force resistive sensors and the temperature sensors are analog. Our weight sensors are wired to pins 0 and 3 and the three temperature sensors are wired to analog pins 1, 2, and 4. The force sensitive resistor required an extra external resistor that served as the range of values detected. We determined that a 1k resistor gave us enough accuracy without being overwhelming. The Arduino Nano BLE we chose to use has both 3.3V and 5V capabilities. The sensors required 3.3V input, so a rail was dedicated for the power source. The same was done for a ground rail. The Bluetooth module utilizes these two rails, with the addition of the RX and TX pins for receiving and transmitting signals.





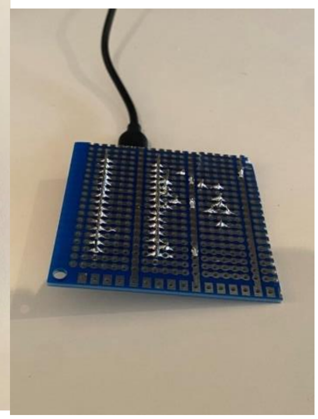


Figure 1: Initial Wiring Figure 2: Protype PCB Figure 3: Soldered PCB

We used the built-in serial monitor to display what values we were getting from the sensors, as seen in figure 4. On the datasheet for the Lilypad sensors there is a conversion formula from the analog data to degrees Celsius. The force resistive sensors did not translate directly to a weight, but through testing described in a later section we determined what values correlated to a child's weight on a car seat. Once these components were confirmed to be working as intended, the device was consolidated into a smaller form using prototype PCB that fits into a 3D printed box as described in the next section in further detail.

```
Weight Sensor1:
617
Weight Sensor2:
52
degrees C1 = 20.58
degrees C2 = 28.32
degrees C3 = 20.58
child: □
Now sending message via BT module...
```

Figure 4: Serial Monitor Output

The third element of the hardware design was the backup battery. While some cars will continue to power the device after the key is removed and the driver exits the car, some cars will automatically stop their chargers from being active while the ignition is off. To cover for any case, we opted for a backup battery in the form of a portable charger. The main feature we needed was that the portable charger does not turn off unless the user decides to manually power it off. Another feature that is important for our device is the battery utilizes pass-through charging. This allows the device to charge while plugged into the car’s cigarette jack while the car is running. Therefore, every time the car's ignition is on, the battery is charging and when the ignition is cut, the battery is able to support our device. After we added this aspect to our design, we wanted to know how many days our device would stay active with only the backup battery to act as its source of power. Using a USB device that tracks the current draw of our device, we found that our device has a battery life of roughly 200 hours when working at full capacity with our device acting as a load. The calculations for this can be found in the design testing section.

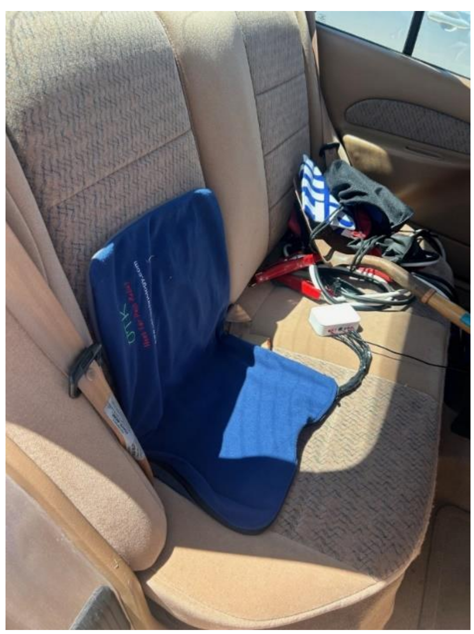


Figure 5: Device in Car

The hardware cost includes the cost of the Bluetooth module, Arduino Nano BLE, 3 Lilypad temperature sensors, 2 force sensitive resistors, 1 prototype PCB board, copper wire, backup battery, and the USB car charger. The total cost of our prototype hardware is $180.85. The price points of each piece are in the Economic Analysis section below. While the price of a single mat may seem steep it is important to note that many of the ways we designed and developed our product are not fully optimized. For example, a sheet of prototype PCB is roughly $10 more than a mass-produced PCB designed for a specific product. Our primary goal was to build a working project before focusing on minimizing costs at every corner.

Hardware: Electronic Housing Design

Before constructing the final housing design of our prototype, there were a few factors that we discussed and took into consideration in order to create the most effective product. The first and highest priority of this entire project was always safety and security for small children, all while giving parents and guardians ease of mind with a simple, user-friendly device. To achieve this, we knew our final prototype had to be easy to install and set up for the user, while also keeping its reliability factor. Taking all of this into account is how we landed on all of our final decisions and design details.

Early on in the design process, we had the idea to use a heating pad mat type of design to house all of our components and a cover for the child to sit on. There were a few different heating pad materials we had to choose from, but we ultimately chose a thin, cotton model because we found that this would be the perfect choice to provide comfort while also allowing the sensors to function correctly without much interference due to the mat material we chose. We also chose this specific mat because it included a zipper, which allows the user to remove the device from the sleeve to clean if they choose, as seen in figure 6.

Figure 6: Device Sleeve

When embarking on this final stage of designing our final product, one of our biggest concerns was the durability of the sensors. Given the nature of small children, we were worried that all of our wires and sensors would not withstand the wear and tear of being used primarily by toddlers. On the other hand, we were also worried that the rigidness of our hardware and equipment might make it uncomfortable for a child to sit on. We did not want to run the risk of a wire or sensor dislodging and poking through the mat causing a child any sort of discomfort. So, in order to combat these concerns, we came to the decision to add a few layers of foam padding on the inside of the heating pad mat. Specifically, we used polyethylene foam sheets that are primarily used for packing. Through our research we found that these were the best choice for us because they have a high heat threshold. The type of sheets we chose are also very thin, which allowed us to layer a few pieces on top of each other to achieve comfort for a child, while also making sure that we still get accurate readings from the sensors.

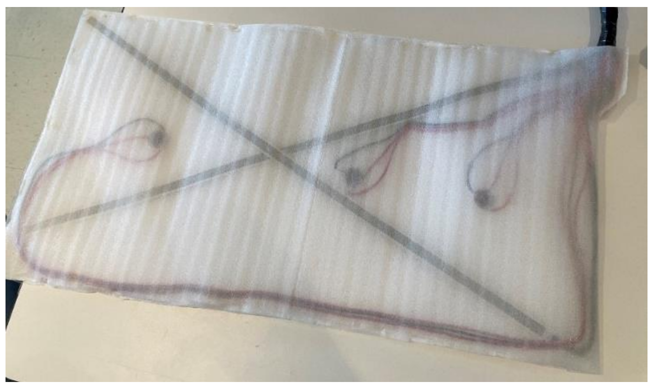


Figure 7: Sensors in Foam

The final factor of our final prototype design that we had to consider was the housing unit for our PCB board that connects our Arduino, Bluetooth module, and all of our sensors onto one consolidated board. While all of these pieces are not very large all together, the casing unit for all of these components was something we had to carefully design because it contains all of the most important components and connections for our entire project. After careful consideration and research, we decided to 3D print a small durable snap box, pictured below in figure 9. We used PLA filament with a high infill percentage in order to create a strong, durable case. We chose a snap box model because it is more difficult to open compared to a sliding lid or a latch box, which were also considered. This makes it child proof and also ensures that all of the components are safe and secure inside this case. The small holes added on the back of the box, seen in figure 8, were added to be able to feed through all the wires that connect all of the sensors in the mat to the PCB board with the Arduino and Bluetooth module. In total, we had to print 3 separate versions of this 3D case until landing on the version we used and were satisfied with. Version one, seen in figure 8, was printed to the correct dimensions needed to fit all of the components inside, however we miscalculated the size of the holes needed to feed the wires through. We corrected this in version two, however we ran into an unforeseen setback with this print that was out of our control. During the print, the nozzle on the printer got clogged which resulted in the print job terminating halfway through, as seen in figure 9. After some slight research, we were able to purge the clogged filament from the nozzle and reprint version two successfully a second time. We then were able to feed through all of the wiring from the sensors and power chord to the PCB board, as seen in figure 10. After successfully placing the PCB board and wires inside the box however, we realized we miscalculated how much room the end of the wires would take up. Rather than undoing all our soldering, wiring, and reprinting a new box base, we decided to draft a new version on the box lid to account for the extra space we needed. For our third and final version of this case, we printed a new lid with more depth to create some more space for all the components in the box, as seen in figure 11.

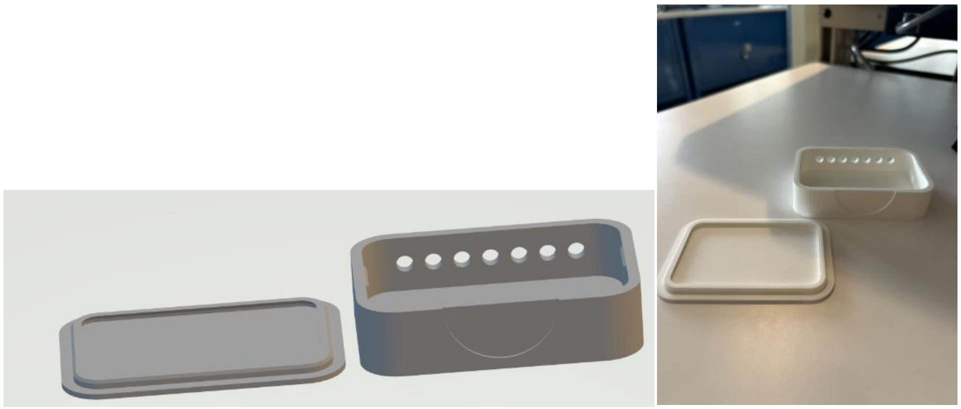


Figure 8: 3D Print Box Version 1

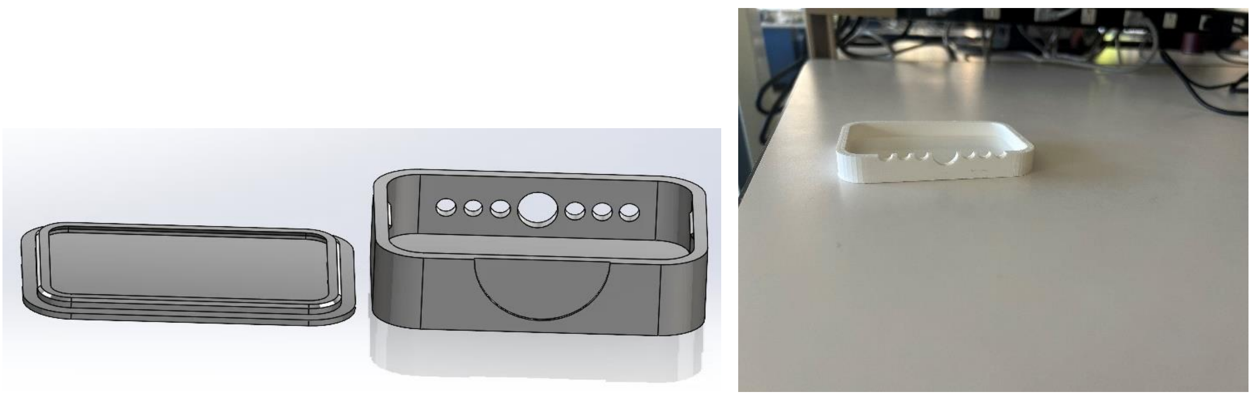


Figure 9: SolidWorks Model

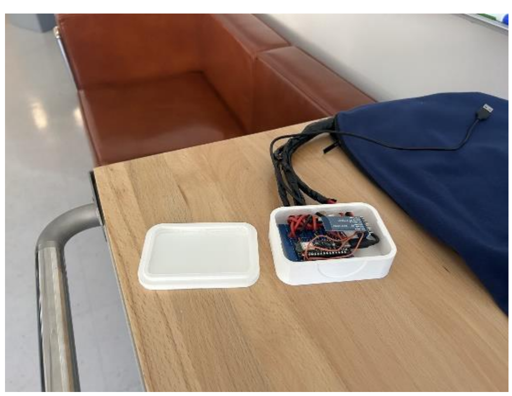


Figure 10:  Wires Fed Through Box

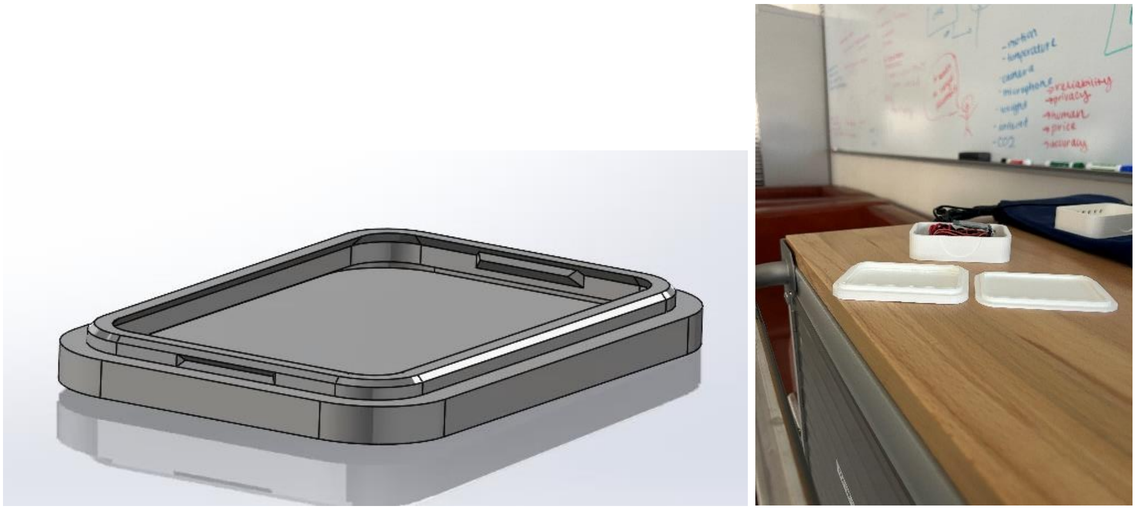


Figure 11:  New Box Cover

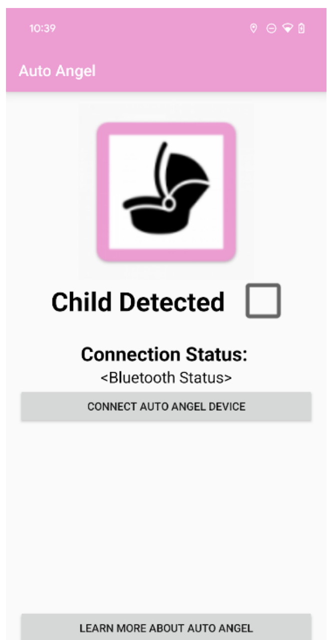



Figure 12: App Interface

The development of the app initially started with a basis of external code examples to build a foundation for the device to connect to an application via Bluetooth functionalities. We then coded the application to implement all the factors necessary to take in the Bluetooth RX buffer received from the external hardware device. Once we could accurately receive the information from the device it would change the interface of the app. We designed the interface of the app to be easily understood and easy to interact with.

The app interface consists of a banner at the top of the page that displays the Auto Angel title. Under the top banner, we displayed the image of our app's logo in the center of the screen. Below the logo, a check box labeled as “Child Detected” is displayed and is unchecked from the start. This checkbox will only mark as checked when both sensors are activated via the external hardware device. Directly below the check box we show the connection status and a button that displays the available devices to connect to. At the base of the page, we have a button that will display a popup in the center of the screen that provides more information about the app and the project as a whole.

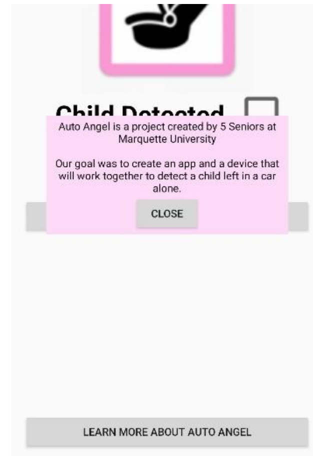



Figure 13: Informational Screen

```
public void run() {
    if (mReadBuffer.getText().toString().trim().equals("1")) {
        sendBluetoothNotification();
        mHandler.obtainMessage(CONNECTING_STATUS, arg1: -1, arg2: -1)
                .sendToTarget();
    } else {
        mHandler.obtainMessage(CONNECTING_STATUS, arg1: -1, arg2: -1)
            .sendToTarget();
    }
}
```

Figure 14: Bluetooth Code

The functionality of the app interface is straightforward with only two functional aspects in order to get the app to run correctly with the device. Once the Auto Angel device is paired to the Android phone through settings it will display under the “Connect Auto Angel Device” button once it is pushed. Once the device is displayed on the app interface you will then click it and the Bluetooth connection will then change to show that the device is connected to the app. The other functional feature of the app is the button on the bottom of the page labeled “Learn More About Auto Angel” once this button is pressed it will display the popup mentioned in the previous paragraph.

Once the device has been paired with the app on the Android phone and the connection status of the device shows a good connection through the app, everything will be fully functional on the app end of the detection of the child in the car. Once a child is detected the checkbox that is displayed below the logo will show a checkmark in the box. This means that the app is connected and detecting a child in the car. Once the device becomes out of the range of the phone a notification will be sent to the phone if there is actively still a child and the checkbox is activated. The setup of the application and the implementation of a notification work together to ensure the notification is sent when it is supposed to be sent. The way this works is in the background of the app through the collection of the information being sent from the hardware device that the app is reading. While the app is receiving the information through an active Bluetooth connection, it knows to display whether or not a child is being detected. Once the phone is out of range of the device it will disconnect and if the last known collected information from the hardware shows there is a child in the car the app will send a notification to the phone. If the phone disconnects from the device and the last known collected information from the hardware shows there is no child in the car then no notification would be sent. Once the phone is out of range of the device, the application will take 5-10 seconds to ensure the phone is out of range before sending the notification.

# Design Testing

When creating a product rigorous testing is necessary to ensure correct functionality, consistent reliability and proper safety. Since our product has to do with child health and safety it was paramount to ensure proper function of AutoAngel. This design testing section encapsulates an overview of the various rigorous tests we went through to evaluate performance and functionality of our system. The tests included temperature and weight sensor calibration, field disconnects, open air disconnects, parking garage disconnects, back up battery longevity, and passthrough charge in a car. The samples of these experiments helped us reaffirm that the product we made is robust and functional.

| Test Name | Part being Tested | Experimental Plan | Measurement Method | Measured Value | Extra Notes |
|---|---|---|---|---|---|
| Field Disconnect (no car) | Bluetooth | In open walkway see how far phone can reach before bluetooth disconnects | Distance(feet) | 178 feet | Open Area outside behind obrein hall of Marquette campus -- many people on cells walking by |
| Disconnect in on top of garage | Bluetooth | Walk away from car, test when/if notification sends | Distance(feet) | 198 feet | Open Area outside of top of parking garage, only a few cars present to block signal |
| Disconnect in Garage | Bluetooth | Walk away from car, test when/if notification sends | Distance(feet) | 165 feet | On a pavement of the Marquette parking garage where multiple cars where present and concrete structures |
| Battery Charges in Car | Battery power source | Plug into car when running, Plug into car when not running. Observe what occurs. | Visual | yes | Ensure when car is on the backup battery is charging via the cigarette port |
| Backup Battery Longevity | Battery Life | Video of mitchell turning off car and battery still working | Visual & Math | 223.33 hours | Measured Current Draw by AutoAngel Device: 30 mA Battery Capacity: 6700 mA Battery Life (in hours) = Current Draw (in mA) ÷ Battery Capacity (in mAh) |

Figure 15: Environment Testing

First, we tested the calibration of our temperature and weight sensors. As spoken about prior, the system has a total of three temperature sensors and two weight sensors to help detect a child. Initial testing was done by pinching each sensor to raise and lower the value. However, this temperature and weight change will be different in the mat in the car with someone sitting on it. When testing in the lab during development we found that pinching the weight sensor gave a value of 600+ resistance and the temperature sensor reached a value of 33 degrees Celsius. Our objective for this experiment was to decide what values the sensors should read and determine a threshold where the device will notify whether a child is present. Once this was complete, we made the necessary changes to Arduino code based on the results. We expected the sensor values to be different once the functional prototype was completed and an actual person would be the testing subject. To run our experiment, we had a team member sit on the AutoAngel mat and collect values of weight and temperature sensors when they stop changing. We repeated the experiment three times to check for consistent values. Our Arduino code displays the temperature in degrees Celsius and weight in resistance Ohms. This data was collected and noted in an excel spreadsheet. We found the average of the values for weight and subtracted 100. The same was done for the temperature sensor but subtracted 2 degrees. Our results: a child should be determined to be detected at 175 on the weight sensor and 30 degrees Celsius on the temperature sensor.

The second test we conducted was examining field disconnect. Up until this point, every minor test had been done in a controlled environment inside a building. For this test, we went outside of O'Brien Hall and set up our testing space on a straight stretch of sidewalk where we could test the device with little interference. The main purpose of this test was to see how far the range for the HC-05 Bluetooth module was when there was no building or car interference. To measure the distance accurately, we used the sidewalk tiles as our measurement tool. Each tile was approximately 2 feet long by 2 feet wide. We set up our device with the HC-05 at the edge of one of the sidewalk tiles and walked away one sidewalk tile at a time continuously monitoring the Bluetooth connectivity using a Bluetooth serial monitor on the connected device. When the stream of messages from the Arduino device stopped sending to the Bluetooth serial monitor, then a disconnect occurred. We predicted that based on the specs of the HC-05, the disconnect would occur between 95 to 100 feet. When conducting our tests we found that after multiple trials the actual disconnect range ended up being around 178 feet, which was better than expected. This was an open-air, outdoor test and the phone that we used to test the disconnect range was a Google Pixel 7. Overall, we considered this test to be a success as a notification was successfully sent to the phone, the disconnect was in an acceptable range, and the overall device worked as needed.

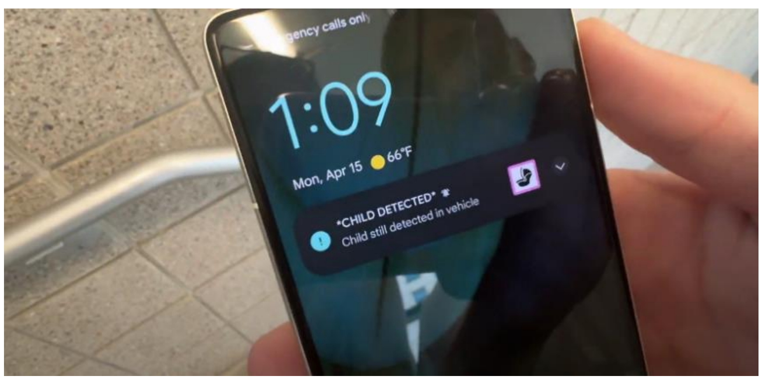



Figure 16: Phone Notification

The third test conducted was observing open air disconnect on top of a parking garage. This test's main goal was to simulate a realistic situation in a setting where the device would be used. We installed the device in one of our team members' cars and drove the car to the top level of the 16$^{th}$ street parking structure on Marquette's campus. There were a few cars parked around the car that we used, but otherwise there were minimal interferences between the device and the phone with our app installed. Our plan was to do a full run through of a realistic scenario with the device in use. First, we placed the sensor mat down in the car, plugged in the device into the car cigarette jack, and verified that the device powered up and connected via Bluetooth to our phone with our AutoAngel app installed. Next, we had one of our team members sit on the mat to simulate that there is a child detected in the car seat and verify that the app receives the signal that there is a child detected in the car. Then, we temporarily shut the car with one of our team members still inside sitting on the mat and had a second team member walk away with the phone until we observe that the Bluetooth is out of range and sends a notification that there is a child still in the car. We used the marked parking spaces as our measuring tool to record the distance between the car device and how far the user got. The spaces were measured to be approximately 9 feet. In total, we were able to get 198 feet away from the car when the Bluetooth disconnected, and we received a notification that there was a child left behind.


Figure 17: In Car Test

The next experiment we ran was to test disconnect in lower levels of the parking garage. When testing our device, we wanted to ensure that even in places where signals can easily be disrupted, everything would still work properly. The process we took to test the device was to park in-between floors in the parking garage between multiple cars. After having someone sit on the device to send the signal to the application we walked through the garage until the phone disconnected from the device. Once the disconnect happened, we could see that a notification appeared on the phone. In diagram, figure 15 shown above, we can see that the measured distance in the parking garage is slightly lower than the distance measured in the open field and on the top of the parking garage. We measured 165 feet from the car inside the parking garage while attempting to create as much interference as possible. We did this by walking between cars and walls to create the worst case inside the parking garage. When the disconnect happens there is approximately a 90 second time frame or less from the time of walking away from the car to when the notification is received on the phone.

The fifth experiment ran was to observe the longevity of our backup battery. Our team's goal for the spring semester was to implement a backup battery to our system to ensure that AutoAngel is always fully operational. All vehicle cigarette outlets work while the car is running, however only some work while the vehicle is turned off. When we realized this was the case, we knew we had to implement a new power source to assure that our device works no matter if the car is on or off. When going about purchasing a backup battery to incorporate into our system we wanted one that had a long battery life and would not automatically turn off. This ultimately makes it easier for the user, as they do not have to charge the backup battery often if the battery life is substantial. This experiment tests the capability of the backup battery's life and its longevity. We tested the backup battery with our device in a powered off vehicle fifteen separate times. Then we let the backup battery run while connected to our Arduino and entire system for 48 hours on end, twice. Test one was implemented first. When the power for the system came from the cigarette outlet, AutoAngel worked great while being powered by the vehicle. Then we changed the power source to be from our backup battery. We turned the car off and plugged the USB into the backup battery and that became the main power source for our AutoAngel system. We ran 15 tests of detecting a child while the backup battery was the main power source, all successful. After running test one, we moved onto going through with our second battery test, testing the longevity of the battery. We had the backup battery plugged into and powering our system over night for two days. The way we were judging the amount of battery loss was through the LED battery life indicator on the product. The whole system was fully on and operational for 48 hours straight, losing minimal power. The backup battery did its job and never failed us. The LED indicator didn't show any signs of battery loss. It was fully operational the entire time and did not show any vulnerabilities or sign of weakness. Test two was successful. The two tests show this backup battery is sufficient and were a necessary addition to our design. We then went through with some calculations to really see how long the battery would stay alive for. We discovered that the battery life is a little over 223 hours. The longevity test was put to the test and we now know that this backup battery will keep our system running for a substantial amount of time when the car is not powered.

Measured Current Draw by AutoAngel Device: 30 mA

Battery Capacity: 6700 mA

Battery Life (in hours) = Current Draw (in mA) ÷ Battery Capacity (in mAh)

6700 ÷ 30 = 223.33 hours.

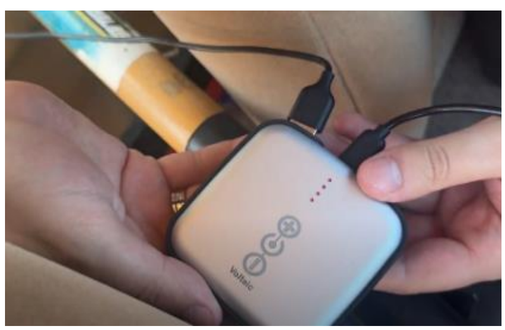


Figure 18: Backup Battery in Car

The last test we ran was examining the passthrough charge in a car. An important aspect to our project was the ability for the device to have supplied power during the time when the car is running and after the car is off. Testing this theory, we can first test the connection of the device while the car is running and then turning the car off and showing that the battery is still supplying power to the device and that the app is showing that the connection is staying active when in range, indicating that the device is on.  When looking at the battery we can see that while the car is running, the battery is being charged. We can also see that the connected device is receiving power despite the battery actively charging, showing that the passthrough charging works as expected.

After all of our extensive testing across various facets of AutoAngel it was with confidence to state that our system has is functional and reliable. Through the calibration of our sensors to real world outdoor simulators and battery power source tests, it can be seen that our new insights validate our design choices. Our experiments displayed to us that our sensors work properly and can detect a human being, a notification is sent when Bluetooth disconnects in open-air, on a parking structure, and in a parking structure. The backup battery is durable, and the power source is reliable. We can conclude that our device works as originally intended. Our disconnect range was larger than we expected. There was little to no difference in the disconnect range when in an open area versus inside a crowded garage. In any case, the user is notified within 90 seconds of walking away from the car. This prevents the child from being left in the car for a long time and allows for the guardian to make a response in a timely manner.

# Economic Analysis

sales forecast (if applicable), component costs (including tooling/production equipment to produce the specific component), production labor costs, and a final estimated product cost. Remember to list all assumptions made when estimating costs (labor rates, production rates, etc., if known) and reference your sources of cost information (vendors, industry sponsors, comparison to existing components, etc.).

Sales Forecast

We have two potential ways to market the product. The first is to sell our product in the current form of our final prototype. The consumer would buy the mat and place it on top of their car seat. The other way would be to sell the technology to a car manufacturer and have this safety feature built into every vehicle. The second option would reduce the overall cost of the device as it would not need a backup battery or separately ordered mat if the system is built into the car's seat and electrical system. Most new cars also have Bluetooth capabilities for music or navigation.

Component Cost

The total component cost breakdown shows the complete cost analysis of every component and any additional items purchased in the process of designing and building our final design. Included with every component listed is the cost per item, quantity, total cost, and justifications, when necessary, with links to the purchase location of each component.

| Product | Price per Item | Quantity | Total | Justification of buying/using product | Link |
|---|---|---|---|---|---|
| Google Pixel 7 | Normally: $399.99 for us: Free | 1 | $0.00 | We wanted to develop our app using Android studio instead of on iOS so we needed a device with Android compatibility that we could quickly troubleshoot apps on | Pixel 7 |
| Arduino Nano 33 BLE | $43.50 | 2 | $87.00 | | Arduino Board |
| HC-05 Bluetooth Module | $10.39 | 1 | $10.39 | | Bluetooth Module |
| Lilypad Temperature Sensors | $5.50 | 4 | $22.00 | | Temp Sensor |

| | | | | | |
|---|---|---|---|---|---|
| Resistive Weight Sensors | $19.95 | 3 | $59.85 | | Weight Sensor |
| USB-C Car Charger | $8.99 | 1 | $8.99 | | USBC car charger |
| Micro-USB Car Charger | $8.99 | 1 | $8.99 | | Micro-USB charger |
| Final Backup Battery | $39.00 | 1 | $39.00 | | Final Battery |
| Velcro | $13.52 | 1 | $13.52 | | Velcro |
| Mat | $19.99 | 1 | $19.99 | | Mat |
| Battery Pack attempt 1 | $7.59 | 1 | $7.59 | | Battery Pack 1 |
| Wire attempt 1 | $12.20 | 1 | $12.20 | | Wire1 |
| Prototype PCB | $11.59 | 1 | $11.59 | | Prototype PCB |
| Foam | $16.45 | 1 | $16.45 | | Foam |
| Final Wire | $10.98 | 1 | $10.98 | | Wire |
| 3D print filament | $32.99 | 1 | $32.99 | | Filament |
| Our print | $3.36 | 1 | $3.36 | It is estimated to cost approximately $0.02-$0.08 per gram of filament and we ran approximately 21 hours of 3D prints. With our printer printing at about 8 grams per hour, the total cost for our prints was about $3.36 on the low-end because our filament cost is also a low-end cost filament. | |

The complete calculated total comes out to be $366.89, and with a given budget of $500, we were successfully able to keep our costs below budget. Since this is the complete cost analysis for every component we did a separate cost analysis of the total cost of only the components used for our final product. This total came out to be $234.17 which is significantly less than the complete cost. These costs could be decreased much more through the use of mass manufacturing and simplification of product design explained later on in the project's legacy with an explanation on the use of PCB boards and wire simplification.

Product labor cost

With our five-person team, each individual contributed a total of 66 hours toward this project. That means a grand total of 330 hours were spent working on the AutoAngel system. In today's economy an entry level engineer is making $37 an hour on average. Taking this statistic and multiplying it by the hours worked, each person would have been compensated $2,442 for their hard work toward AutoAngel's development. That number multiplied by five – for the five teammates – totals $12,210.00. Over twelve thousand dollars would have been dealt out in production labor costs to compensate our group. Take into consideration that this is using the baseline entry-level engineer's average per-hour pay rate. A lot of hours, dedication, and hard work was put into the development of our product. This dollar amount accurately portrays how valuable each team member was to this given project.

Final estimated product cost

| Product | Price | Quantity Bought | Quantity Used in product | Math | Went into Product Price |
|---|---|---|---|---|---|
| Arduino Nano 33 BLE | $43.50 | 1 | 1 | n/a | $43.50 |
| HC-05 Bluetooth Module | $10.39 | 1 | 1 | n/a | $10.39 |
| Lilypad Temperature Sensors | $5.50 | 4 | 3 | $5.50*3 | $16.50 |
| Resistive Weight Sensors | $19.95 | 3 | 2 | $19.95*2 | $39.90 |
| Micro-USB Car Charger | $8.99 | 1 | 1 | n/a | $8.99 |
| Final Backup Battery | $39.00 | 1 | 1 | n/a | $39.00 |
| Velcro | $13.52 | 5 feet | 2.5 feet | $13.52/2 | $6.76 |
| Mat | $19.99 | 1 | 1 | n/a | $19.99 |
| Prototype PCB | $11.59 | 3 pack, used ¼ of 1 | 1/12 | $11.59/12 | $0.97 |
| Foam | $16.45 | 50 sheets, used 3 | 3/50 | $16.45/25 | $3.90 |
| Final Wire | $10.98 | 20 feet | 15 feet | $10.98*0.75 | $8.24 |
| Our print | $3.36 | 1 | 1 | n/a | $3.36 |

**1 Unit Total = $201.50**

# Design Risk Analysis

The key process in the design project outlines what goals must be met before the project can move forward. One of the first key steps is to decide on what sensors, microcontroller board, and any other parts must be ordered. The delivery status must be monitored closely to ensure safe delivery. Upon arrival, the parts must be tested and verified that they are functioning properly with the board itself. After that, data collection and early testing begins. The parts must be calibrated, and the output understood. Once these hardware tasks have been fulfilled, the software tasks begin. The program must first accurately detect when a child has been left in the seat and the parent walked away. Once the program can do this, the program must communicate with the user and notify them with an alert. The final key process is to develop a simple application that works with the Bluetooth connection on the device to also notify the user on their phone.

During the design process complications are likely to occur. The first major risk is with part ordering. To minimize the risk of parts not arriving on time, the team member who placed the order should keep track of the order and estimated arrival time on a regular basis. Multiple sources for parts should be considered in case of supply chain issues. The next issue is with software or hardware incompatibility. If this occurs, other versions of the part should be ready to order or if the issue is much larger, other versions of the product must be considered. Our primary idea is to use load cells and Bluetooth communication. If either of those prove to be unusable, we have other designs prepped. Hardware is liable to fail or break for any number of reasons. Backup parts should be ready to minimize downtime in case of a hardware failure. Any coding project is subject to unexpected bugs. The most efficient way to avoid complete code failure is to incrementally test the code so any issues are caught early on rather than deep in the design process. Below is a simple version of the risk assessment.

| Event | Severity | Probablity | Risk | Mitigation Plan |
|---|---|---|---|---|
| Parts not delivered/delayed | 4 | 30% | 1.2 | Team member that orders the parts keeps track of shipping process. Order from multiple sources. |
| Parts not compatible | 4 | 5% | 0.2 | Review datasheets thoroughly before ordering. |
| Parts break | 1 | 10% | 0.1 | Backup parts ordered immediately and ready on standby |
| Complete prototype failure | 5 | 20% | 1 | Other prototype designs kept around until a working prototype is completed. |
| Application software failure | 2 | 50% | 1 | Routine testing of code to prevent early bugs causing catastrophic errors later. |

Figure 19: Original Risk Assessment

These risks were all successfully mitigated with our final design. In the section below, Project Legacy, the final paragraph outlines potential additions to the project that could improve upon the final design. To see more details, reference that paragraph. With new additions come new risks. We have identified the two largest issues that could come with the new additions. The first is related to contacting authorities in the event of a missed notification. If emergency services are called to the location for a false positive detection, it could prevent them from being located at a real emergency. The only way to mitigate this risk is to do rigorous testing to make sure the device is as accurate as possible. There would be no time for emergency services to double check if the emergency is real or not, so near perfect accuracy is the best fix. The second large issue would be with a device that is adaptable to multiple environments. Temperatures can fluctuate wildly daily, even hourly, in extreme cases. The solution to this once again comes down to rigorous testing. The code would need to track how long the temperature change takes to occur.

## **Project Legacy**

At every step of product development, we researched potential solutions and their viability for our needs and timeline. We learned through both this research and making mistakes along the way. For both Android Studio and Arduino code, the internet proved very abundant with example projects and resources to work through any roadblocks we encountered. We learned that Apple has significantly stricter policies when it comes to accepting unknown Bluetooth devices. In addition to this, it is way easier to load a personally developed app onto an Android device. For testing purposes, this saved us a significant amount of time and headache. This is ultimately why Android proved to be the better option for short-term app development.

There were design choices that worked well, and we would not change given the chance to start again. The Lilypad temperature sensors, force sensitive resistors, and HC05 Bluetooth module proved to be highly effective components. The temperature sensors were accurate within half a degree Celsius and the HC05 range exceeded our expectations for the disconnect range. After assembling the prototype and soldering all the wires onto the final device all the sensors worked flawlessly. This made the calibration of the sensors in the car particularly easy.

In any project there are design choices that are functional, but not optimal. One of our final hurdles in the product design was how we were going to place the sensors into a car seat and feed the wires out into the housing that holds the Bluetooth module and Arduino device. We ultimately decided to use a mix of hot glue and staples to secure the wires to the thin foam inside the sleeve. This functions as a temporary solution, but in the long run those would come loose and could result in false negatives.

For any future teams working on a similar project, we have outlined further improvements that can be made. Our product has a lot of bulky wires soldered into a prototype PCB board. Given more time, a more compact solution would be to design PCB board specific to this project using software like KiCad that only requires the sensors to be soldered to it. One of product goals was for the mat in which the sensors are laid in to be discrete and comfortable. This was achieved; however, thinner wires are always an improvement. Our App was designed for an Android device. A future group could take our code and modify it for the App store and Apple devices. This would make it accessible to virtually anyone with a cellular device. Another feature that could be added is to automatically contact emergency services and a trusted number if the app user does not interact with notification that a child is still in the back seat. A final addition that we thought could be added to the device is tuning it for multiple environments. Our temperature sensors are currently tuned for a Wisconsin spring, but in theory the device would need to be functional in a wide range of resting temperatures. This could be done by tracking temperature change rather than setting a threshold.

# Appendix

To view all code documentation, reference our github repository.

https://github.com/benminick/AutoAngel?tab=readme-ov-file

# References


[1] “Force sensitive resistor (FSR),” Adafruit Learning System, https://learn.adafruit.com/force-sensitive-resistor-fsr/using-an-fsr (accessed Jan. 15, 2024).

[2] “Simple Android bluetooth application with Arduino example,” mcuhq, http://mcuhq.com/27/simple-android-bluetooth-application-with-arduino-example (accessed May 1, 2024).

[3] "Unattended Children in Vehicle State Laws," KidsinCars: https://www.kidsandcars.org/laws/unattended-children-in-vehicle. [Accessed: April 29, 2024].

[4] Glassdoor. (2024, May). "Milwaukee Entry Level Software Engineer Salary." Retrieved from: https://www.glassdoor.com/Salaries/milwaukee-entry-level-software-engineer-salary-SRCH_IL.0,9_IM564_KO10,39.htm

[5] C. Ababei, Mess lab, http://www.dejazzer.com/software.html (accessed Jan. 2, 2024).