11/14/24 Day 1:

Today I was planning on making the UI system work. This system uses 5 push buttons, an LED, and a 4 digit 7 segment display. We didn't get the 7 segment display yet but I tried to get the other components done. Today I connected the Dev Board and buttons to the breadboard. I used the Serial monitor to check that the button inputs work when pressed. I used a capacitor and a resistor in order to account for debouncing. All 5 button inputs were recorded.

11/15/24 Day 2:

Today I planned on creating the timing of the clock. I used the Serial Monitor on the Arduino IDE in order to show the time since the 7 segment display was unavailable. I coded the buttons to change the time, stop the alarm, change the alarm time, and set the alarm to be off. I didn't know which button to use in order to change the actual time so I decided that pressing both the minute and hour buttons at the same time would start changing the actual time. I added an LED to the circuit that shows whether the alarm is on or not. If the LED is off and the alarm time is reached there will not be a response from the device. I also added a speaker to the circuit. The speaker I used runs using the tone() function in Arduino IDE. I produced a sound frequency of 3000 hz using the speaker. I also realized that the debouncing circuit wasn't working, so I decided to use code instead. Even if the code detects that the button was pressed more than once in a 0.3 second span it will not execute more than once.

11/18/24 Day 3:

Today I planned on integrating the Ultrasonic Sensor to the circuit. This works by measuring the time between a pulse being received and a pulse being sent and using that to determine the distance from the nearest object. We don't have the motors yet so I implemented a function in the code called randomTurn(). This function randomly calls rightTurn() or leftTurn(). Since there are no motors I put a print statement saying "turn right" or "turn left". This has the side effect of constantly calling randomTurn() since we don't stop this process while we turn since we aren't actually turning. This will be fixed once we get the motors.

12/2/24 Day 4:

Today I planned on adding the 7 segment display and motors to the device. Over thanksgiving break I was able to obtain the 7 segment display, motors, and wheels. I first added the 7 segment display to the device. I was using print statements to show the time on the display so I added the libraries to the code for the display and replaced these print statements. After doing this the code failed to upload. I had believed that this occurred because of the libraries that I had installed but the issue persisted after I removed them. After doing some research it turned out that the display being connected to the microcontroller during the upload/startup procedure causes issues with the flash memory, preventing the device from starting up or uploading code. By removing the power cable during startup/upload from the display I was able to run the code, and by reconnecting it after the device starts up it works correctly. One issue though, is that since it is not connected during startup the display will not display anything until a minute passes. Since I have set the default start time to be 1:00:50, it only takes 10

seconds to display something but I would still prefer to have it work on startup. This is a lower priority issue though, as it is not a big issue to wait 10 seconds for the time to display and we can see through the LED that the device is operational. I also moved the buttons onto a separate breadboard. This will be on top of the housing and operate as a sort of UI panel. If we were to improve this design we would want to have a UI panel that hides the wires and is not a breadboard but for the purposes of showcasing the functionality this is a functional UI panel. I obtained some n-type MOSFETs in order to control the motor logic but the gate voltage was too high for the dev board to control the motor logic. Tomorrow I plan on acquiring either a motor driver or MOSFETs that have a lower gate voltage.

12/3/24 Day 5:

Today I planned on moving the UI breadboard to the top of the housing, placing the other breadboard inside of the housing, and making the motors operational. I was able to acquire a motor driver today. This allowed me to control the logic of the motors with the dev board. I disconnected the UI bread board from the main breadboard and placed the main breadboard inside of the housing unit, then I placed the UI board on top of the housing and reconnected them. It was difficult to press the buttons before this as the wires were in the way but this placed all of the wires beneath the breadboard, preventing them from being in the way of the user's hand. Then I connected the motors to the drivers and tested that their direction was correct. Then I attached the wheels and tested it. The product is completely functional, although the dev board sometimes has a loose connection. Sometimes the dev board has to be pushed back into the breadboard

as some of the pins don’t connect properly. This tends to make the speaker not make a sound during the alarm but this is fixed by pushing down the dev board. We also replaced the speaker with a different model that has a higher decibel value. This model has a decibel value within our required range of 75 dB to 85 dB with only one speaker. The original speaker required 4 speakers in order to reach this range. With all of this completed the device has been completely assembled and works properly now.