Deliverable H: Prototype III and Customer Feedback

GNG 1103 Group 12C

Names:

Abdullah Abdulmajeed

Alyssa Wang

Antonia Zupu

Spencer Henry

Yomna Elsahili

Date
March 29th, 2020

To: Prof. David Knox, Ms. Rani Damuluri, Mr. Selameab Demilew

This report will detail the final results of our prototype testing progress and will serve to exhibit our LifeLine device in its comprehensive form.

Introduction

The goal of this deliverable was to combine all the subsystems we had previously tested separately in prototype 2 into a final comprehensive prototype that could demonstrate the integrated functions of our device. Ideally, this deliverable would have shown a complete, functioning, aesthetically-pleasing device that we would have shown on Design Day, but due to unforeseen circumstances that is not possible as we lost access to many necessary resources.

Instead, we will display all of the progress we have made up until this point, as well as highlight the steps we would have completed if the COVID-19 outbreak had not occurred. Similarly to the past deliverables, this prototype will be broken down into its 4 components: the phone application, device frame, overdose detection, and circuit. The phone application will consist of explaining the decisions made while designing our user interface. The device frame will explain the changes made with regard to its shape and closing mechanism as we refine the final device. The overdose detection describes how the failsafe will function in terms of the triggering conditions and response protocol. Lastly, the circuit aspect will describe the progress and issues we encountered when trying to include the battery, switch, and charging port.

After combining all of the aspects and forming a complete conceptual idea of our final device, we seeked customer feedback in order to help gauge how people perceive our device. Moreover, there will be an updated BOM chart included to clarify the total expenses made.

Plan For Deliverable H:

Prototype	Description	Person(s)	Done
Implementing an on/off switch and battery/battery charger to the device.	Connecting them to circuit and assuring its functionality.	Abdullah / Antonia	Yes
Setting up the connection between the HC-05 and arduino	Adding to the arduino code so that all blood-oxygen readings are sent to the app via the HC-05.	Alyssa / Yomna / Spencer	Yes
Testing Comfort with Device Frame	Will make sure the device isn't intrusive or uncomfortable.	Antonia	No
Setting up failsafe and alarm	Designing the app so that it asks the user to confirm the overdose and setting off an alarm if overdose is confirmed.	Spencer / Yomna / Alyssa	Yes
Device Assembly	After the previous are done, the last step will be to finalize the building.	All	No

Final Status Update:

As of this moment, we have completed the majority of tasks that were expected to be done for prototype 3 and our final device. We have successfully created a functional circuit that reads pulse oximeter values and can connect through bluetooth to our phone application. Once the phone application is connected, there will be updated values for the pulse and spO2 levels on the device which could trigger a failsafe alarm and send the user's location to an emergency contact if an overdose is detected. This device would have been worn on the wrist such that the oximeter sensor could be flush against the user's skin on the top of the wrist. Once the product is bought, the user must also download the *LifeLine* app on their phone and connect it to the bluetooth module, HC-05, such that it can retrieve the values from the main wristwatch device. From there on, the user is only responsible for wearing the device on a daily basis while ensuring they charge it and turn it on when they intend to use opioids.

There have only been a few aspects of our device that we were unable to implement: the battery, charging port, switch, and wrist strap. However, the details regarding these aspects will be discussed within the future improvements section of this deliverable. The following image will provide a visual of how our device should have looked like, had we been able to compact all the components into the device frame and purchase a watch strap.

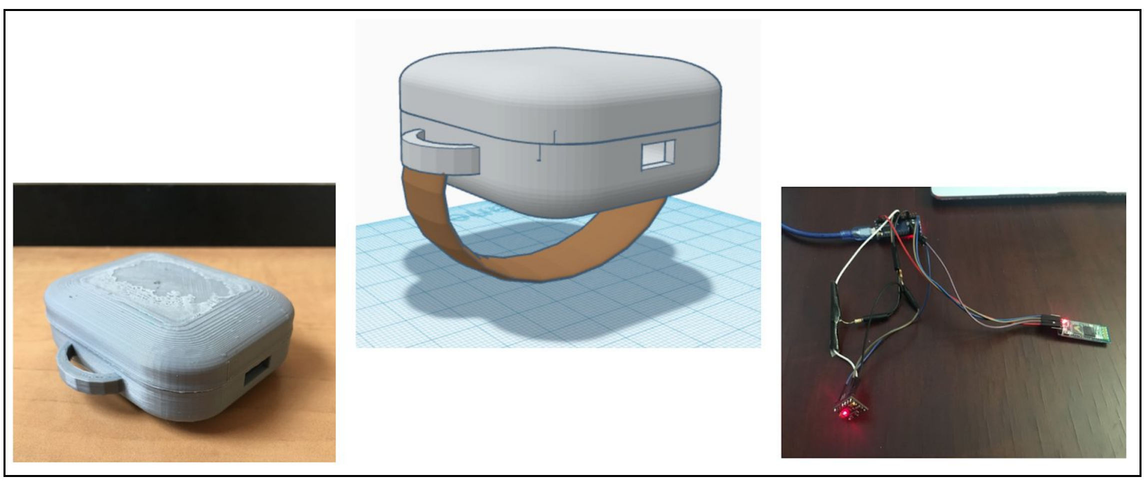


Figure 1

The complete functionality of this device will be demonstrated in the following video: (Please have your volume turned on for the video).

Video Demonstration: https://youtu.be/VFOqZ02ltkY

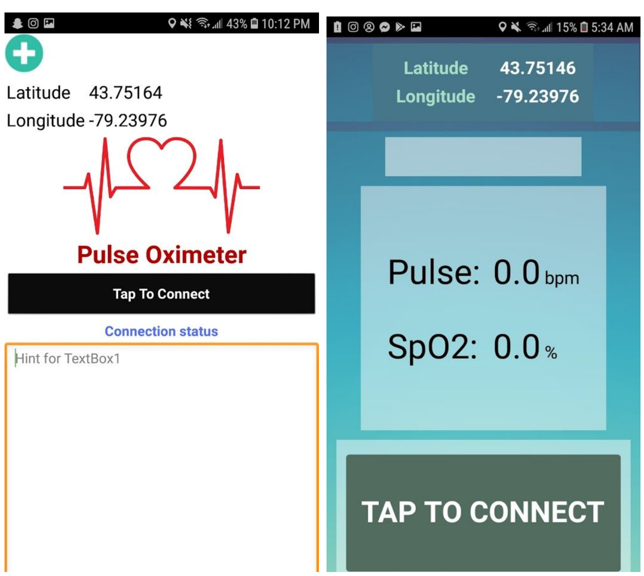
1. The LifeLine App:

After reviewing the quality of work we had done for the second prototype, we realized a lot of things that we were still missing about our application and about the design and aesthetic of our app.

First, we were missing very important features. We still didn't have a working failsafe and alarm and also a working automatic text feature (How these features work will be discussed in parts 3 and 4 of our deliverable. For this section we will focus on the visual design of that area). We had not finished up filling in the instructions we had wanted for the CPR and naloxone screens and were running very little on time when it came to finalizing these features.

Secondly, the look of our application needed serious improvements. It was extremely basic and boring, not pleasing to the eye, and was not easy to navigate around as the text in certain areas was small, buttons were tiny, and features that were more important than others weren't highlighted or shown in a way to fit that important role. So, the goal of our final prototype was to get as much done as possible within the time limit we had (beginning of the second prototype to the final presentation) by figuring out the features and designs we wanted to complete first, and by getting more members to assist with the large load we had.

Oximeter (Main screen)



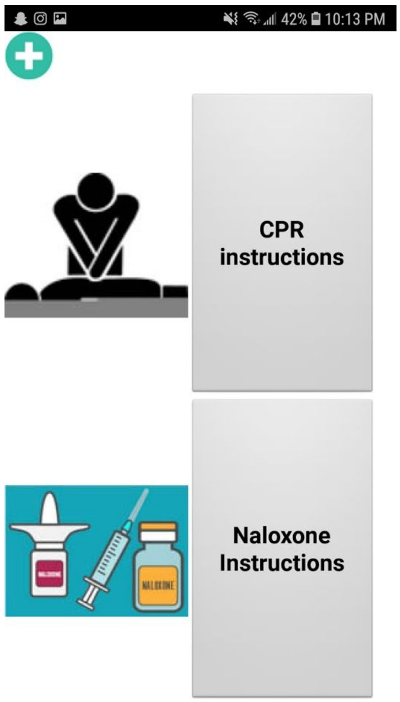
Figures 1.1 (prototype 2 design) and 1.2 (prototype 3 design)

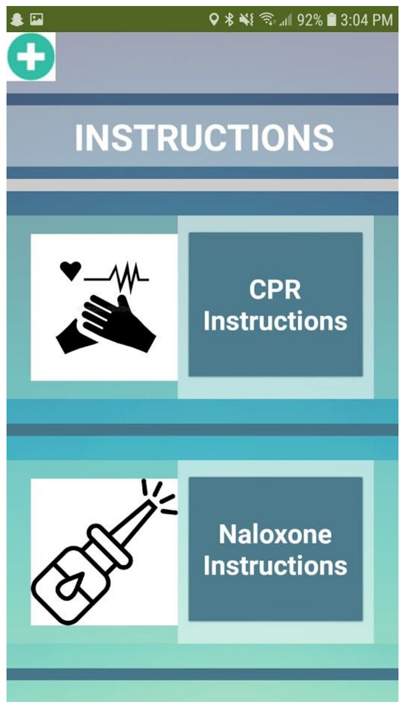
The major changes we made to the redesign of this feature are:

- Less white space, cleaner more organized look with resized buttons and images

- Included teal/bluish palate, new design
- Larger text being printed is much easier to read, complete redesign of the screen helped highlight the areas that were more important and that users should pay attention to
- Easier on the eyes for users

Instructions Screen





Figures 1.3 (prototype 2 design) and 1.4 (prototype 3 design)

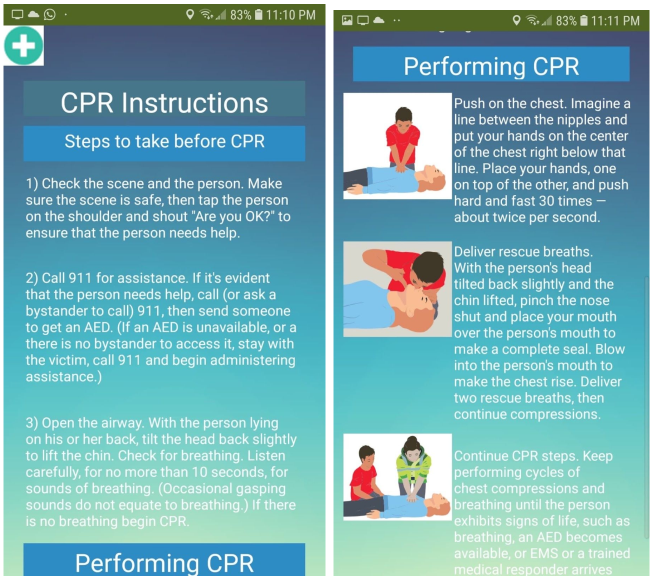


Figure 1.5 (a), (b)

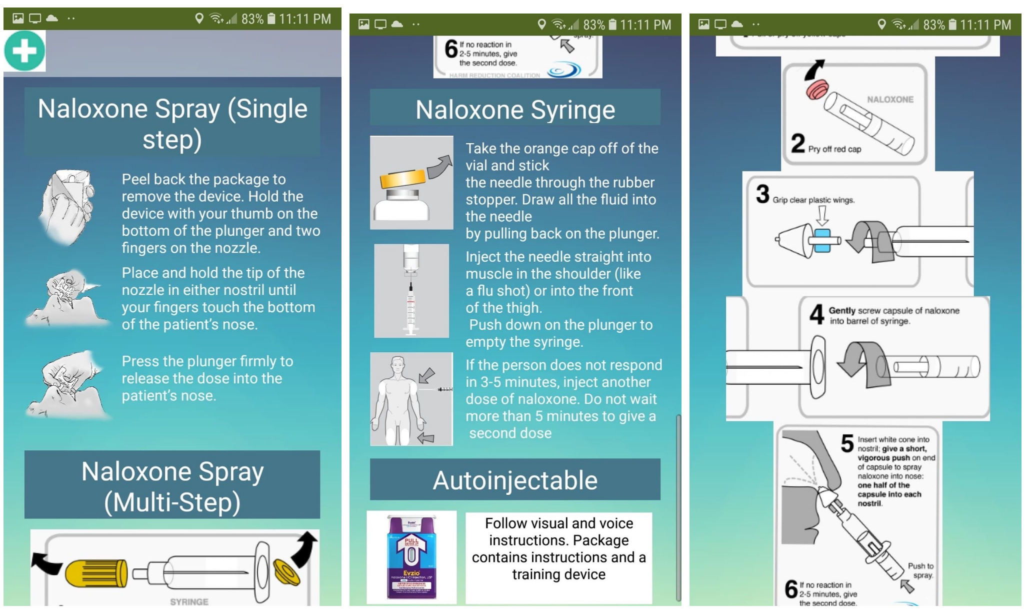
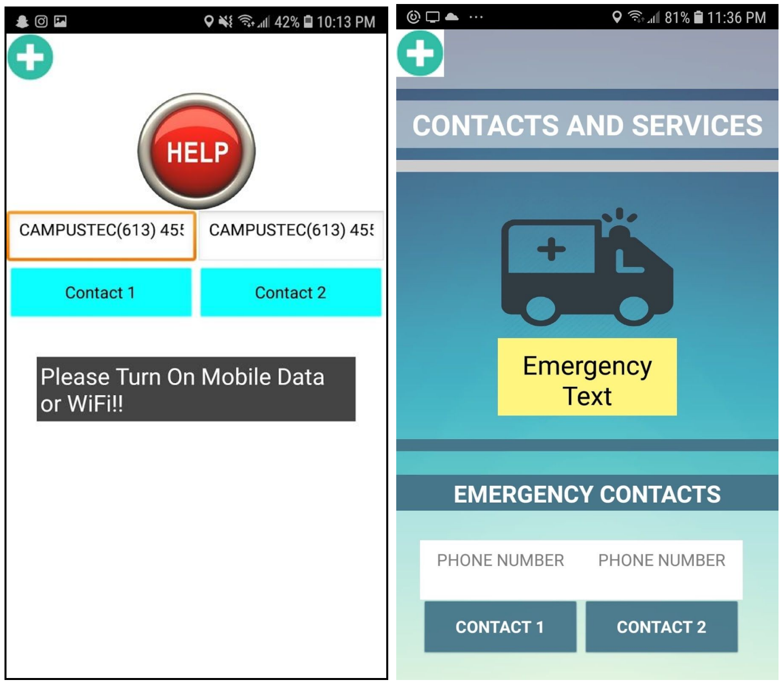


Figure 1.6 (a), (b), (c)

The major changes we made to the redesign of this feature include:

- Less white space, cleaner look with resized buttons and images
- Included teal/bluish palate, new design
- Included step by step instructions

Location/Automatic Text Setup Screen



Figures 1.7 (prototype 2 design) and 1.6 (prototype 3 design)

The major changes we made to the redesign of this feature include:

- Larger icons for visibility and navigation
- Buttons are more spaced out, less white space
- Much cleaner design, incorporated teal/bluish background
- Help button is more recognizable, larger and easier to understand

With the complete redesign we were able to do, we finished the task of making our application more visually appealing with the massive changes we made for navigation, simplicity and attractiveness. We kept certain aspects such as our toggle menu, and improved on others such as the redesign of our oximeter reader, and finished making and adding the final features that we wanted to include into our application.

2. The Device Frame:

For this week's prototype, we had three goals in mind. We focused on finalizing the placement of the charger and sensor hole, adding a hole for the switch, increasing the size to fit the device components, and lastly including the MAX30100 grooves we previously had.

Instead of gluing the device frame shut, like stated in the previous deliverable, we had to figure out a way to keep the device frame components accessible so we are able to fix it if any components disconnect. With this in mind, we started considering hinges.

Figure 2.1(a) and *Figure 2.1(b)* would be much too bulky so we decided against it.

Figure 2.1(a),(b),(c)

Our first design for this prototype as seen in *Figure 2.1(c)*, includes holes on the top of the lid of the device and cylinders on the bottom so that the lid can slide into the bottom and stay in place.

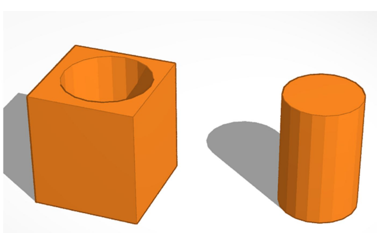


Figure 2.2

However, after testing out this closing mechanism by 3D printing out a rectangle with a cylindrical hole and the cylinder to fill the hole as seen in *Figure 2.2*. we deemed this not possible since we had great difficulty properly sizing the hole such that it fit the cylinder perfectly, not too tight that it would not fit and not too loose that it would not stay in place. Thus, we concluded that the designs in *Figure 2.1(a)*, *(b),* and *(c)* would not work and they follow the same principles.

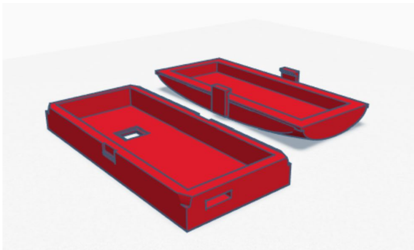
 

Figure 2.3 (a), (b)

We realized that our clasps were too detailed for the printer, and wouldn’t work in keeping the device closed, or even slightly clasping on. Thus, we had to abandon this idea. At this point, the COVID-19 pandemic has started to become more serious, so we were beginning to get more comfortable with the idea of just gluing our device shut as we knew that time was limited. However, we had one last idea that we wanted to try.

Figure 2.4

After researching more designs online, we came across this idea where a thin layer of material connected the lid and bottom, such that they could possibly be folded like a book. This wouldn't necessarily keep the device completely shut but it may keep the pieces held together. We determined that the 3D printers in the makerspace at UOttawa use PLA, which was similar to what we saw online, and thus assumed it would work. Unfortunately, we had a lot of difficulties printing as many printers kept breaking or were in use, and when we finally managed to print one, the thin layer completely broke when we tried to bend it.

Finally, we lost hope on the closing mechanics of the device as classes were getting cancelled and we knew there was only a couple days until makerspace would close, so we just wanted to perfect our holes, finalize our grooves and increase the size. We managed to achieve that, however, this device frame was still quite bulky and had sharp edges, which we didn’t like for the discreet aspect of the device. So then moving forward, we tried to focus on remodelling our device frame to make it more aesthetically pleasing.

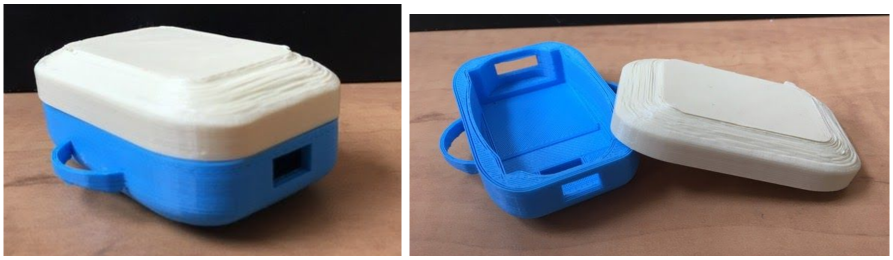


Figure 2.5

This device had a much nicer look and feel, however, was too small for our components and battery to fit. This was because we forgot to take into account that rounded edges would result in less space within the frame. So our next step was to make it bigger, and fix the size of the holes and they lost precision in this print, and as well as the grooves.

Just as we were ready to print the next, and hopefully finally device frame, makerspace closed and all we had was the design for the next device frame on tinkerCAD. Luckily, one of our team members had a friend who had access to a 3D printer, however, we could only use it a limited amount of times as we did not want to consume all of their PLA material.

We were able to come up with this final device frame, which had perfect hole sizes, and grooves for the sensor as well as the newly included sleek edges for the aesthetics of the device.

Figure 2.6. Final Device Frame

The only flaw was that this device was slightly larger than necessary, and so with more time we could have printed a smaller device to make it even more discreet and comfortable for the user. Unfortunately, we could not print another device since we had all gone home at this point and no longer had access to any printers. If makerspace was still open we may have printed a last device as well as filled it down to be smooth and not have any pieces of PLA sticking out.

3. The Overdose Detection:

For this prototype we focused on putting all the functional elements of the app together, such as the failsafe and the alarm. Initially, we thought it would be best to evaluate the blood oxygen level within the arduino code and use an if statement to assess whether the blood oxygen reading is a safe one or not. If not, the arduino code would send the reading to the app as usual, but with a warning as well to trigger the alarm protocol on the app. We did not end up going with this approach as it involved introducing several variables and complicated the arduino code.

Instead, we moved the danger assessment to be done in the app and left the arduino code in its simplest form possible. The arduino code will get blood oxygen level readings from the MAX, and send them to the application. The application displays these levels, but also assesses them to ensure they are above 90%. If the application finds the readings to be less than 90%, then the alarm protocol is triggered.

Alarm Protocol:

Once the application detects a blood oxygen level less than 90%, a timed notification appears asking if the user is okay (shown in *Figure 3.1*). There is only one button that says “Yes” and if the user presses it, the alarm protocol is disabled and the application functions normally. If no response is detected within 30 seconds, then the application takes this as confirmation that an overdose is detected. A loud alarm will start blaring and a GPS location of the user will be sent to the two emergency contacts preset.

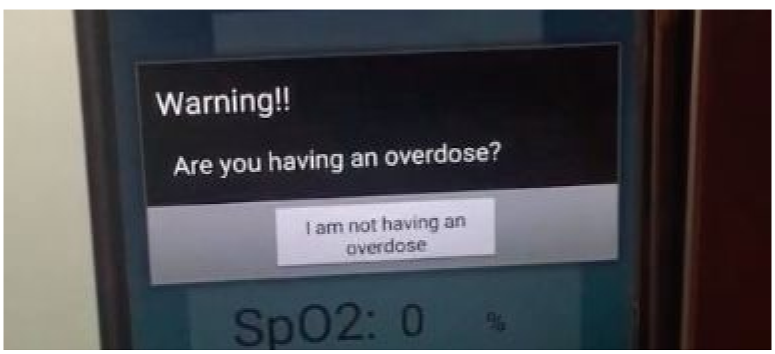


Figure 3.1. Failsafe-Alarm Notification

Stopping Criteria:

We tested this by stimulating an overdose by taking the finger off of the sensor to get a reading less than 90%. We observed that the notification came up, and when we didn’t press anything for 30 seconds, the alarm went off and the text message was sent. We tried it again, but this time clicked “Yes” when the notification popped up and the alarm protocol was cancelled and the application went on displaying blood oxygen level readings.

4. Circuit Changes:

The circuit for Prototype III is essentially made up of two parts.

First, the inner circuit of the device itself that we tried to take off the breadboard to fit inside the device frame. The circuit is not fixed to anything and is connected through wires only. Resistors are also connected across these wires to supply power to SDA and SCL pins on the MAX30100 chip. The oximeter is fully functional and so is the HC-05 bluetooth communication module if the Arduino is supplied with sufficient power through a USB drive or a 5V battery.

Second, the outer circuit that is made up of the battery, powerboost 1000C, and power switch. The powerboost we received seems to be defective, as it would heat up when connected to anything so we decided to not risk connecting it. As a result, we were unable to include the battery into the circuit as it only produced 3.7v which is not enough power to run the circuit without the powerboost micro booster. Therefore, the outer circuit is not functional. We also have a power switch that would have been connected in series to the power wire coming out the powerboost and into the Arduino. The power switch would break the circuit and shut off the current going into the Arduino. It would be there so that the user can fully turn off the device throughout the day when the device is not in use.

Planned Execution under Normal Conditions:

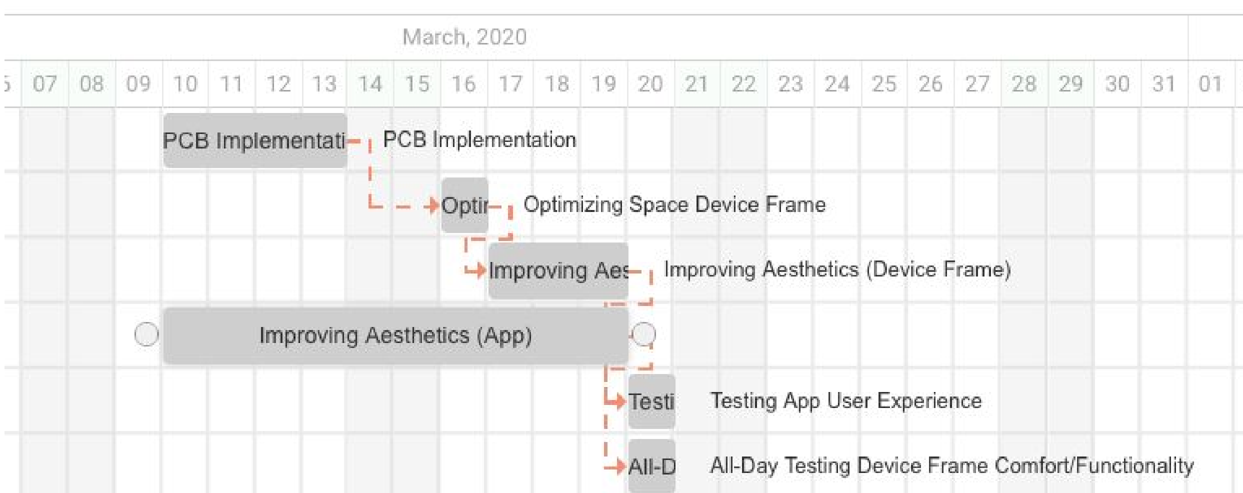
Due to unforeseen circumstances, we were not able to complete our initial schedule. We had planned to dedicate the last two weeks to assembly and aesthetics, and then testing. For assembly, we had hoped to solder on the components onto the PCB board instead of having them cluttered inside the device frame for stability and reliability. This was not possible, however, due to the closing of CEED facilities. We were able to get our hands on a solder, but did not have a desoldering tool and did not want to take any unnecessary risks.

Because of the lack of PCB implementation, we resorted to soldering together the wires to get the circuit off the breadboard. During this process, the battery and powerboost 1000C (battery charger) that we ordered came in, but when we attempted connecting them to the rest of the circuit, the powerboost started overheating and so we decided it would be best to keep them disconnected. With more time and resources, we would have been able to replace, or possibly fix, the defective powerboost. This was unfortunate as we could not complete our final circuit. Ideally, we would have been able to build our final circuit onto the PCB board and align it inside the device frame to show a fully functioning device. We also decided against ordering a watch strap, considering the circumstances, and so that would have been something we'd have done.

We also would have dedicated time to improve the aesthetics of the device frame and the watch strap to make it more visually appealing. We considered filing down the frame and possibly painting it to give it a slicker look. Also, the current size of the device frame is estimated to be too big, and so we would have spent time trying to optimize the size in makerspace after the cricut was built onto the PCB board. We would have had more time to work on the aesthetics of the MIT app as well. We would have liked to add extra features, such as giving the user the option to pre-set his emergency blood-oxygen level reading (which we have decided on 90%) and they have the option of choosing their failsafe time before the alarm protocol starts (that we have as 30 seconds).

Lastly, we were not able to complete any form of testing on our prototype. Initially we would have liked to dedicate at least a full day of testing to it. We would've gotten someone to wear the device for twenty-four hours and see how it holds up. From this test we would be able to see several things. It would tell us if the failsafe was activated too often, which means the pulse oximeter was not in direct contact to the wrist at all times, indicating that the watch strap or device frame grooves might need to be reconsidered. It would also give us user feedback on the comfort of the device and whether it was discrete enough that the day passed without getting anyone's attention. It would also test if the device is durable enough for everyday activities, as we did not consider making it waterproof. There are a lot of uncertainties, such as whether readings can be taken if the area of contact is sweaty or wet, that we could have eliminated if we had the time to test. These tests could not take place as our device was not compacted in its device frame.

Gantt Chart:



In the end, we would have been able to document our findings from our testing and deem whether our final prototype met our preset specification from Deliverable C:

Design Specifications	Relation	Value	Units	Verification Method
Functional Requirements				
Long-Lasting battery	>	24	hours	Test
Fast detection	<	3	minutes	Test
Fail-safe detection	=	yes	N/A	Test
Constraints				
Cost	≅	100	$	Test
Weight	<	170	g	Test
Non-invasive	=	yes	N/A	Test
Non-functional requirements				
Aesthetics (discrete, simple)	=	yes	N/A	Test
Durability	=	yes	N/A	Test
Product Life	>	1	year	Test

Customer Feedback:

Following the most completed version of our device, we sought out feedback to better understand the audience's opinion of our device as well as see where we could make improvements. However, we were limited in our ability to gain feedback due to the COVID-19 outbreak, especially with respect to the quality of work that would be displayed for judgement. This means that getting opinions directly from potential users was not possible. Nevertheless, we tried our best to gain critical responses through our classmates and friends. This was executed in a few different methods.

We had downloaded the app on our phone and allowed our peers to play around with it. By doing this, we were able to directly test the quality of the user experience from each person since we observed the ease of navigating through the app and quality of our user interface. We received lots of positive responses regarding the ease of usage, visuals and colour scheme. However, one thing that was suggested for improvement was a simpler instructions page. This is because in the case of an emergency, having to read in depth instructions for CPR or naloxone distribution seems inconvenient as it reduces the amount of time available for executing the instructions and may confuse the reader.

After our final presentation, we received feedback from our classmates and professor. Some of our classmates made positive comments regarding the development of our device frame and phone application. We were glad to hear that there were improvements made when comparing our initial model to our final device. Professor Knox had also been quite pleased with the progress made with the phone application, however, he inquired about the on and off switch, similarly to what was discussed with our client during our pitching presentation. Specifically, it was asked if including the switch was reasonable to request for the user to remember before using drugs. The implementation of this switch was something that created great debate, however, we were able to come to the conclusion of including it. This was because we hoped that since the user is willing to buy this product for their safety, they would be able to take one more additional step to ensure they are getting the best service. If we were able to implement the switch, we would be able to take readings at closer intervals which would ultimately increase the accuracy of the readings and decrease the chances of the user having a false alarm. Our client had understood our reasoning and was willing to give it a chance.

Video Demonstration: https://www.youtube.com/watch?v=plCwYbiTDkc

Conclusion:

This deliverable demonstrates the final progress on the development of LifeLine. Currently the device is separated into three physical components. The aforementioned outer circuit and inner circuit. Lastly the third part of this prototype is the device frame. 3D printed and made from plastic, it's big enough to fit the components of the outer and inner circuits. It has holes aligned to fit the necessary ports. However, since the outer circuit is not functional and the device is not rechargeable, we did not want to seal the device within its frame yet.

The phone app is fully functional and is almost seamlessly synchronized with the oximeter readings. In the event of an overdose where the oximeter will record low oxygen saturation. The app will go into panic mode. It will sound an alarm to notify nearby people to help the user. It will also automatically send a warning text to a specific contact with the location of the overdose. Furthermore, a failsafe measure was also implemented in case of a false positive. When oximetry measurements fall low, a pop up will show up so that the user can disable the alarm. If the user fails to disable the alarm within the designated time. Panic response will initiate.

Development of the device stalled before completion in the last weeks since under the pretext of the COVID-19 crisis CEED was closed. CEED facilities provided services and equipment that was imperative to us at that point. Furthermore, we received the defective powerboost chip and couldn't replace it because of reasons pertaining to the crisis.

Under normal circumstances had we been able to access the resources we needed. We could've run the process to completion in terms of replacing the defective chip, putting all the components together and finally performing the final testing stages as outlined previously.

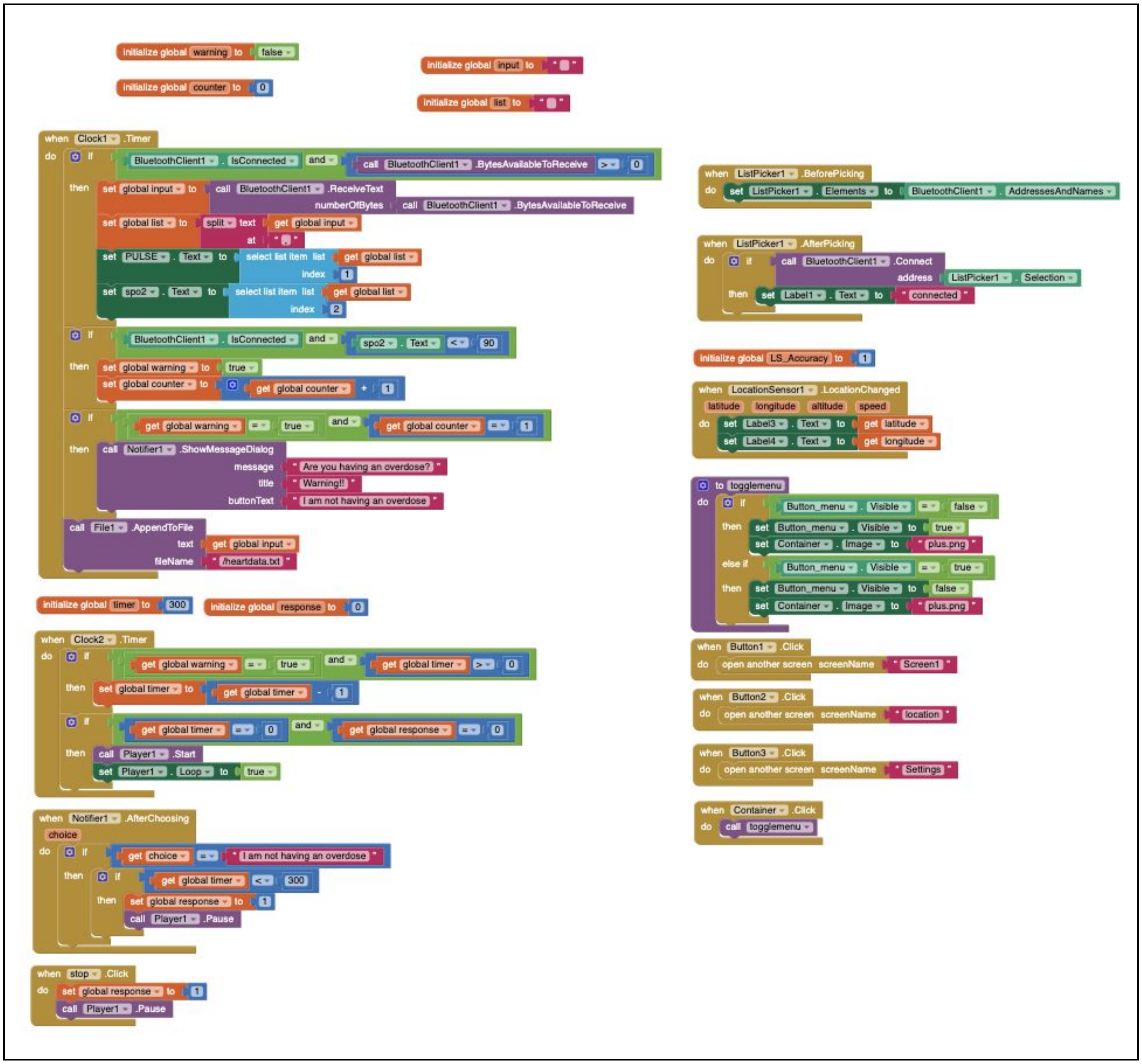
Appendix:

Final Code

```
#include <Wire.h>
#include "MAX30100_PulseOximeter.h"
#include <SoftwareSerial.h>

SoftwareSerial lifeLine(0, 1);
#define REPORTING_PERIOD_MS     1000
PulseOximeter pox;

uint32_t tsLastReport = 0;

void setup()
{
    Serial.begin(9600);
    lifeLine.begin(9600);
    if (!pox.begin()) {
        for(;;); }
    else {
      }
}
void loop()
{
  int pulseox;
    pox.update();

    if (millis() - tsLastReport > REPORTING_PERIOD_MS)
    {
        Serial.print(pox.getHeartRate());
        Serial.print(", ");
        Serial.println(pulseox=pox.getSpO2());
        if(lifeLine.available())
        lifeLine.write(pulseox);
        tsLastReport = millis();
    }
}
```

Final MIT Code Blocks

Home Screen (code for bluetooth connection and alarm failsafe)


initialize global warning to false
initialize global counter to 0
initialize global input to " "
initialize global list to " "
when Clock1 .Timer
do if BluetoothClient1 . IsConnected and call BluetoothClient1 .BytesAvailableToReceive > 0
then set global input to call BluetoothClient1 .ReceiveText numberOfBytes call BluetoothClient1 .BytesAvailableToReceive
set global list to split text get global input at " "
set PULSE . Text to select list item list get global list index 1
set spo2 . Text to select list item list get global list index 2
if BluetoothClient1 . IsConnected and spo2 . Text < 90
then set global warning to true
set global counter to get global counter + 1
if get global warning = true and get global counter = 1
then call Notifier1 .ShowMessageDialog
message " Are you having an overdose? "
title " Warning!! "
buttonText " I am not having an overdose "
call File1 .AppendToFile
text get global input
fileName " /heartdata.txt "
initialize global timer to 300
initialize global response to 0
when Clock2 .Timer
do if get global warning = true and get global timer > 0
then set global timer to get global timer - 1
if get global timer = 0 and get global response = 0
then call Player1 .Start
set Player1 . Loop to true
when Notifier1 .AfterChoosing
choice
do if get choice = " I am not having an overdose "
then if get global timer < 300
then set global response to 1
call Player1 .Pause
when stop .Click
do set global response to 1
call Player1 .Pause
when ListPicker1 .BeforePicking
do set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames
when ListPicker1 .AfterPicking
do if call BluetoothClient1 .Connect
address ListPicker1 . Selection
then set Label1 . Text to " connected "
initialize global LS_Accuracy to 1
when LocationSensor1 .LocationChanged
latitude longitude altitude speed
do set Label3 . Text to get latitude
set Label4 . Text to get longitude
to togglemenu
do if Button_menu . Visible = false
then set Button_menu . Visible to true
set Container . Image to " plus.png "
else if Button_menu . Visible = true
then set Button_menu . Visible to false
set Container . Image to " plus.png "
when Button1 .Click
do open another screen screenName " Screen1 "
when Button2 .Click
do open another screen screenName " location "
when Button3 .Click
do open another screen screenName " Settings "
when Container .Click
do call togglemenu

Final BOM

Option #1 (If MakerStore Return is possible)

BILL OF MATERIALS			
Material	Price (CAD$)	Option #	Bought Or Not
Pulse Oximeter	$24.00	1	Bought (2+$12)
Arduino	$7.00	5	Bought
Bluetooth Module	$11.00	2	Bought
attery Charger + Batter	$43.09	1	Bought (Roboshop)
Wires	$2.44	1	Bought
Total Estimated Price:	$87.53		

Option #2 (If return is not possible)

BILL OF MATERIALS			
Material	Price (CAD$)	Option #	Bought Or Not
Pulse Oximeter	$24.00	1	Bought (2+$12)
Arduino	$7.00	5	Bought
Bluetooth Module	$11.00	2	Bought
MakerStore Charger	$12.00	1	Bought (want to return)
attery Charger + Batter	$43.09	1	Bought (Roboshop)
Wires	$2.44	1	Bought
Total Estimated Price:	$99.53		