Lifelines Breathalyzer

Eugene Oliver, Ryan Do, Adriene Almacen

March 28th, 2017

Lifelines: Portable Breathalyzer Project

Project Website: EugeneHasJeans.github.io

Declaration of Joint Authorship

The team "Designated Drivers" which consists of Eugene Oliver, Ryan Do, and Adriene Almacen confirms that the project of the Lifelines Breathalyzer is a combined group effort and is a combination of our own thoughts and ideas. The work of this entire project was split as equally as possible. Eugene Oliver worked with calibrating the sensors and was in charge of hardware design and working with the app layouts. Ryan Do worked with the database in terms of setting it up, connecting it with the app and maintaining it. Adriene Almacen was in charge of mobile application design and maintenance. The distribution of testing the hardware and software for bugs and issues are discussed by the three of us and worked on all together. Before any changes were done on the project, a group consensus had to be made. All work that was used for guidance and help has been acknowledged and cited in the reference area of this report.

Approved Proposal

Proposal for the development of Lifelines: Portable Breathalyzer

Prepared by Eugene Oliver, Ryan Do, Adriene Almacen

Computer Engineering Technology Students

https://eugenehasjeans.github.io/

Executive Summary

As students in the Computer Engineering Technology program, we will be integrating the knowledge and skills we have learned from our program into this Internet of Things themed capstone project. This proposal requests the approval to build the hardware portion that will connect to a database as well as to a mobile device application. The internet connected hardware will include a custom PCB with sensors and actuators for testing and finding a user's alcohol level. The database will store information on past alcohol levels of the user, and information on the legal alcohol levels in the province/state. The mobile device functionality will include information, based on the found alcohol level, of friends or taxis to call, GPS location to find the closest hotel and other features. This will be further detailed in the mobile application proposal. We plan to collaborate with the HRT – Hospitality, Recreation and Tourism department here at Humber college.

The hardware was completed in CENG 317 Hardware Production Techniques independently and the application will be completed in CENG 319 Software Project. These will be integrated together in the subsequent term in CENG 355 Computer Systems Project as a member of a 3 student group.

Background

The problem solved by project is the frequency of people that drink and drive and endanger their lives and the lives of others. Using the alcohol level tester, people can realize how much alcohol is actually in their system and from that can create a solution for their safety instead of going into a bad path. Every year there is a call of help for people to stop drinking and driving, but yet every year there are accidents that always lead to a common problem of alcohol consumption. Studies have shown that out of all the

young drinking drivers who cause harm on the road, the largest age group is 19 years of age, which is also a big age group for the use of technology and phones/gadgets. We believe that the more people who use this project will lower the rate of this problem and bring us one step closer to a solution for drinking and driving.

We have searched for prior art via Humber's IEEE subscription selecting "My Subscribed Content" and have found and read three which provides insight into similar efforts.

This first journal that was found talks about developing safety measures to prevent and detect impaired driving.
(Sakairi, 2012)

The second journal talks about how to create a more precise heart rate reading using formulas. (Brüser et al., 2015)

The last journal explains and talks about a certain sensor that measures the alcohol concentration in a human.(Venugopal et al., 2008)

In the Computer Engineering Technology program we have learned about the following topics from the respective relevant courses:

- Java Docs from CENG 212 Programming Techniques In Java,

- Construction of circuits from CENG 215 Digital And Interfacing Systems,

- Rapid application development and Gantt charts from CENG 216 Intro to Software Engineering,

- Micro computing from CENG 252 Embedded Systems,

- SQL from CENG 254 Database With Java,

- Web access of databases from CENG 256 Internet Scripting; and, Wireless protocols such as 802.11 from TECH152 Telecom Networks.

This knowledge and skill set will enable us to build the subsystems and integrate them together as my capstone project.

Methodology

This proposal is assigned in the first week of class and is due at the beginning of class in the second week of the fall semester. My coursework will focus on the first two of the 3 phases of this project:

Phase 1 Hardware build.

Phase 2 System integration.

Phase 3 Demonstration to future employers.

Phase 1 Hardware build

The hardware build will be completed in the fall term. It will fit within the CENG Project maximum dimensions of 12 13/16" x 6" x 2 7/8" (32.5cm x 15.25cm x 7.25cm) which represents the space below the tray in the parts kit. The highest AC voltage that will be used is 16Vrms from a wall adaptor from which +/- 15V or as high as 45 VDC can be obtained. Maximum power consumption will be 20 Watts.

Phase 2 System integration

The system integration will be completed in the fall term.

Phase 3 Demonstration to future employers

This project will showcase the knowledge and skills that we have learned to potential employers.

The tables below provide rough effort and non-labour estimates respectively for each phase. A Gantt chart will be added by week 3 to provide more project schedule details and a more complete budget will be added by week 4. It is important to start tasks as soon as possible to be able to meet deadlines.

Labour Estimates	Hrs	Notes
Phase 1		
a) Writing proposal.	9	Tech identification quiz.
b) Creating project schedule. Initial project team meeting.	9	Proposal due.
c) Creating budget. Status Meeting.	9	Project Schedule due.

d) Acquiring components and writing progress report.	9	Budget due.
e) Mechanical assembly and writing progress report. Status Meeting.	9	Progress Report due (components acquired milestone).
f) PCB fabrication.	9	Progress Report due (Mechanical Assembly milestone).
g) Interface wiring, Placard design, Status Meeting.	9	PCB Due (power up milestone).
h) Preparing for demonstration.	9	Placard due.
i) Writing progress report and demonstrating project.	9	Progress Report due (Demonstrations at Open House Saturday, November 7, 2015 from 10 a.m. - 2 p.m.).
j) Editing build video.	9	Peer grading of demonstrations due.
k) Incorporation of feedback from demonstration and writing progress report. Status Meeting.	9	30 second build video due.
l) Practice presentations	9	Progress Report due.
m) 1st round of Presentations,	9	Presentation PowerPoint file due.
n) 2nd round of Presentations	9	Build instructions up due.
o) Project videos, Status Meeting.	9	30 second script due.
Phase 1 Total	135	
Phase 2		
a) Meet with collaborators	9	Status Meeting
b) Initial integration.	9	Progress Report
c) Meet with collaborators	9	Status Meeting
d) Testing.	9	Progress Report
e) Meet with collaborators	9	Status Meeting
f) Meet with collaborators	9	Status Meeting
g) Incorporation of feedback.	9	Progress Report

h) Meet with collaborators	9	Status Meeting
i) Testing.	9	Progress Report
j) Meet with collaborators	9	Status Meeting
k) Prepare for demonstration.	9	Progress Report
l) Complete presentation.	9	Demonstration at Open House Saturday, April 9, 2016 10 a.m. to 2 p.m.
m) Complete final report.	9	Presentation PowerPoint file due.
n) Write video script. 2nd round of Presentations, delivery of project.	9	Final written report including final budget and record of expenditures, covering both this semester and the previous semester.
o) Project videos.	9	Video script due
Phase 2 Total	135	
Phase 3		
a) Interviews	TBD	
Phase 3 Total	TBD	

Material Estimates	Cost	Notes
Phase 1		
Raspberry Pi 3 Starter Kit	$119.99	(Canakit) Amazon –B01CCF9BYG
Electronics Parts Kit	$119.99	Humber – SKU #163
MQ-3 Sensor	$11.95	Amazon.ca – B01ISMV6G8
XD-58C Sensor	$18.99	Amazon.ca – B01AUVMFIS
Solder Kit	˜ $40.00	Humber
Soldering Iron	˜ $20.00	Humber
Phase 1 Total	˜ $330.92	
Phase 2		
a) Materials to improve		
Phase 2 Total	TBD	

Phase 3

a) Off campus colocation	<$100.00
Shipping	TBD
Tax	TBD
Duty	TBD
Phase 3 Total	TBD

Concluding remarks

This proposal presents a plan for providing an IoT solution for the abnormal amount of people that dangerously drink and drive above the alcohol limit. This is an opportunity to integrate the knowledge and skills developed in our program to create a collaborative IoT capstone project demonstrating my ability to learn how to support projects. I request approval of this project.

Abstract

The purpose of this breathalyzer is to help prevent drinking and driving. As drunk driving is one of the leading causes of accidents on the road. This breathalyzer can help fix the issue of drunk driving by determining the users blood alcohol content (BAC) as well as their beats per minute (BPM), so that users can determine beforehand if they are able to drive home safely. When the alcohol or heart rate test is taken, data is retrieved from the sensors and stored on a database. In addition, the data is also displayed on an android application along with the date and time of when the test was taken. Furthermore, the android application will aid the user by allowing them to determine if it is safe to drive and will be given options such as calling a friend, calling a taxi or seeking the nearest hotels in the area.

Table of Contents

Illustrations and Diagrams

Figure 1: System Diagram for Lifelines Breathalyzer

This figure is a system diagram of the project, describing general structure of how the Lifelines: Breathalyzer works.

Figure 2: Board layout for custom PCB This figure is a schematic for the custom PCB we used. This shows the layout of the PCB to make sure everything looks okay.

1. Introduction

Alcohol and driving under the influence continues to become a major problem in today's society and every year, many lives are taken due to accidents caused by drunk driving. Most people do not own a breathalyzer and many bars are not required to carry one due to the high cost. The work described in this technical report was undertaken because we wanted to try to create a solution where you can track your Blood Alcohol Concentration (BAC) as well as your heart rate and present the user with options depending on what range their BAC falls under. Our breathalyzer connects to an android phone where our mobile app displays readings and stores them into a database providing past history records to lookup in the future. The mobile app provides user accounts to be able to synchronize data across multiple devices and retrieve data from the database to display the average values for the BAC and heart rate. After the user provides a breath sample and a reading shows up, a choice is presented to the user to either call for a taxi, call a friend, or find a hotel nearby.The objective of this breathalyzer is to reduce the amount of drunk drivers that danger themselves and others while on the road. When working with the breathalyzer we ran into multiple problems. One of the problems was creating a case that is small enough to bring around easily, but big enough to hold all the components. This was fixed by creating a smaller case compared to our original one. Another problem was having to find a way to power the Raspberry Pi without using the AC adapter. The solution to this, was finding a small 5V portable charger that has a micro-usb connection. A unique approach in this project was seeing the best way to calibrate both sensors. We realized that the heart rate sensor works best when there is a black surface underneath it. The alcohol sensor has a potentiometer that can adjust the sensitivity of the readings.

2. System Requirements Specifications

2.1 Technology Introduction

2.1.1 Purpose

The purpose of our Lifelines: Breathalyzer is to make sure people are off the roads if they are intoxicated or not safe to drive. Drinking and Driving is a huge issue in our world today, and by using this technology to check if there is alcohol is someone's body, we can keep lives out of danger.

2.1.2 Product Overview

This product is built using with a custom made PCB board, Raspberry Pi and two main sensors. The MQ-3 Gas sensor is what we use when to check if there is alcohol in your system. The XD-58C is what we use in our product to check your heart rate.

2.1.3 Targeted Audience Group

Our targeted audience group would obviously be people over the legal drinking age. This device would mainly be used for personal reasons, but in occasion you can bring it with you if you were going to a party, or hosting one to make sure everyone is safe to be behind a wheel.

2.2 Product Information

2.2.1 Main Functionality

The main functionality of the product is to test a user's Blood Alcohol Content (BAC) and determining if they are okay to drive home or not. If not, then the user will be given a number of options such as calling a friend, calling a taxi, or finding a hotel/motel to stay at for the night. The user can also measure their heart rate as well to see if there is a correlation between their BAC and heart rate.

2.2.2 Extra Requirements

This breathalyzer uses a rechargeable battery that allows it to be portable. No internet connection is needed only Bluetooth connection. This connection allows the device to connect to a user's mobile device.

2.2.3 Best Performance

For best performance out of this technology you can consider a few things. To get the most accurate reading the heart rate sensor should have a dark colour behind it, because of the LED on the sensor it works better with dark areas. Also, try not to move as much as possible because when steady you get more steady readings.

2.3 Overall Description

2.3.1 Database

We will be using Google's Firebase mobile and web application platform for user authentication on logins and our real-time database. Users will be able to sign up for an account and login to access their personal previous alcohol and heart rate results, as well as the averages for both blood alcohol content and heart rate. The database will be a JSON database which will contain the user's unique ID (UID) as well as user information that is taken from when the user signs up for an account. Their UID is automatically generated by Firebase Auth when they sign up for an account. Under their UID, holds their name, address, city, phone number, province, and licence type. When they perform an alcohol test or heart rate test, the resulting data is stored under their UID. The data is then fetched and displayed showing the last 5 results and the averages when they navigate to the past results page in the mobile application. (Developed by Ryan Do)

2.3.2 Hardware

The hardware portion of our project is a breathalyzer. This breathalyzer will be used to help eliminate drunk driving by giving the user the ability to test their blood alcohol content (BAC) level as well as their

beat per minute (BPM). The device tells the user if it is safe to drive home or gives the user alternative options if their unable to drive. The device uses two sensors that help determine the users BAC and BPM; it uses an alcohol gas sensor and heart rate sensor. The sensors are connected to a Raspberry Pi that runs a program that gathers a reading from the sensor. The device itself is inside a small acrylic case that holds the contents inside as well as a small rechargeable battery to make the device portable. Data from the sensor are also sent to a connected device and displayed on an application as well as sent to a database. (Developed by Adriene Almacen)

2.3.3 Mobile Application

For our project, there will be a mobile android application which will be able to show data collected from our Lifelines: Breathalyzer. Once the application is open you must sign up and log in to be able to see past results which were read from our two sensors on our hardware, MQ-3 Gas Sensor and XD-58C. Without logging in/signing up, the past results button would be hidden, which leaves a button to start the alcohol test or heartbeat test. After these tests, you will be brought to a results page that presents the readings. If the reading from the alcohol test detects alcohol in your body, it will give you options to take such as calling a friend, calling a taxi or seeing hotels close by to you. The last option you would have is to look at past results (if you are logged in). This is where the data is taken from the database. The user will be able to look results stored in the database which will show them their past BAC and BPM readings that were done previously, along with their average BAC and BPM readings. If you are logged in, once you run a test it would be added and updated to the database. (Developed by Eugene Oliver)

2.4 Future Considerations

2.4.1 Operating Environment

To be able to use our product, you would need an android smartphone or tablet that is on android API 19 (Android 4.4 KitKat) or above.

2.4.2 Safety Considerations

In order to maintain safety when using this technology you must keep some things into consideration:

- Make sure to keep sanitation in mind when passing the breathalyzer around

- Do not expose the technology to any liquids to prevent errors in accuracy or malfunctions

- 5 Volts DC is what should be used when providing power to the Lifelines: Breathalyzer

- Be sure not to tamper with the technology while it is powered on

2.4.3 Future Additions

Before the end of the term, we plan on designing and cutting out a new acrylic case to house the Raspberry Pi and the sensors we need. We plan on trying to make this as portable as we possibly can, given the resources that are provided.

2.5 Build Instructions

2.5.1 Introduction

This section will give you the main instructions and provide help if you want to create your own Lifelines: Breathalyzer. The Designated Drivers, a group consisting of Eugene Oliver, Ryan Do, and Adriene Almacen chose to create this project because we feel like this is a good step to take in order to fix the problem of drinking and driving. Instead of people thinking they're okay to drive with alcohol in their body, they can use our project to see if there is a great amount of alcohol in their body along with seeing their heart rate. It's a cheap alternative, and isn't too hard to make yourself!

Using a gas and a heartbeat sensor, users can plug those into a PCB and use a raspberry pi to display the readings on either a computer or phone when the app is ready and completed!

You can have your very own Lifelines: Breathalyzer product just by following this section. The only thing that would keep you from doing it much faster would be the delivery dates on items. After putting in the hard work, you can use it to impress people with and hopefully use it yourself to keep you far away

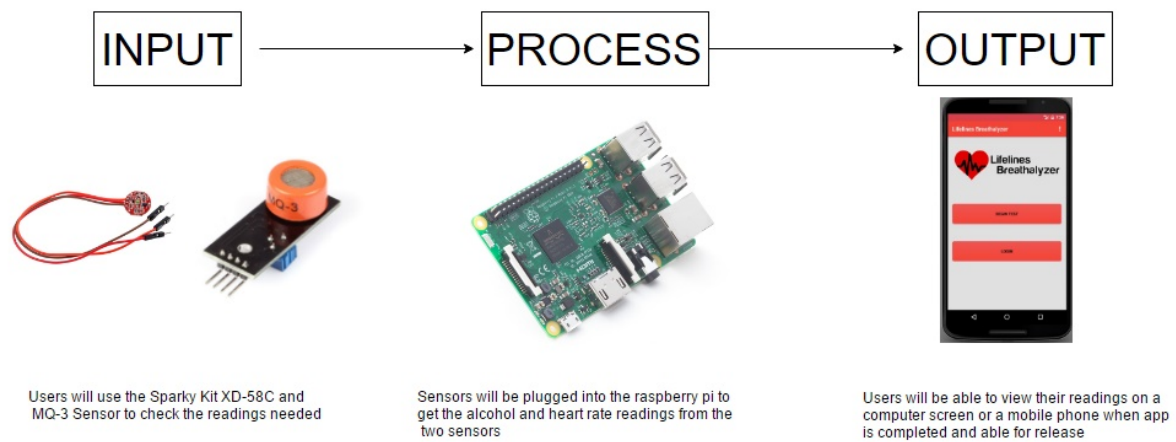from drunk driving. You yourself can help reduce the problem of drinking and driving just by making this.

Figure 1: System Diagram for Lifelines: Breathalyzer

2.5.2 Bill of Materials

Required parts and materials for this project are shown below. Most of these parts are pretty cheap which makes this project not that expensive, but that is because we already had a raspberry pi, electronic parts kit, and the PCB kit was paid for as a part of our tuition. It is important to mention that these prices do vary considering where you get them, you don't need to buy the exact same parts as us.

Item	Quantity	Cost	Supplier & Part Number
Raspberry Pi 3 Starter Kit	1	$119.99	(Canakit) Amazon - B01CCF9BYG
MQ-3 Sensor	1	$11.95	Amazon.ca - B01ISMV6G8
XD-58C Sensor	1	$18.99	(JutaTech) Amazon.ca - B01AUVMFIS
Electronics Parts Kit	1	$119.99	Humber - SKU #163
Jumper Wires (120 pack)	1	$19.99	(Elegoo) Amazon.ca - B01EV70C78
TP-Link 5200mAh Charger	1	$15.99	Canada Computers - (TL-PB5200)
Solder Kit	1	~$40.00	Humber
Soldering Iron	1	~$20.00	Humber
Power Cables/Connectors	1	included	Humber/Amazon

Again, these are just the parts and prices for the things we bought. Prices may change over time, but our total comes to around $370.

2.5.3 Time Commitment

Lifelines: Breathalyzer is a project that we worked on in our last year in Computer Engineering Technology. In the 5th semester of this program a lot of work had to be juggled with other courses, which is why as a whole it took about 15 weeks to complete everything. If you work on this continuously with no other tasks in your way, it shouldn't take that long considering you do everything correctly! In this chart below, we break down how much time was taken on each main task of the project.

Thing To Be Done	Time Taken To Complete (Approx.)
Looking for and Purchasing Parts + Delivery	1.5 hours + 2 weeks
Assembling case and setting up Raspberry Pi	1 hour
Editing your custom PCB	30 minutes
Soldering/Testing/Troubleshooting your PCB	3 hours
Creating a case for your project	1 hour
Testing/calibrating the sensors	2 hours
Setting up the project	5 minutes

After breaking down the parts of the project, it is pretty easy to tell that it's not a very time consuming project to complete. If you are very committed to this project then it shouldn't be very difficult to complete this in these time frames.

2.5.4 Mechanical Assembly and Setup

To keep things simple, for mechanical assembly we will need to break up the parts into main sections. First we will talk about how you have to set up the raspberry Pi, what needed to be done in order for it to work properly. Next we will talk about creating your own PCB, and soldering it. After that we will talk about connecting the parts together and powering it up. Lets get started!

Raspberry Pi

These steps might vary if you get different parts, my Raspberry Pi starter kit included a lot of things to make the process a lot easier.Once you receive your Raspberry Pi, you can start by connecting a keyboard and mouse to the USB ports. Next go ahead and push in the SD card included, and use the HDMI connector and connect it to a monitor so you will be able to see the display. Lastly, plug in the power adapter to turn on the raspberry pi. Things will flash on the screen and text will flood a black terminal, but eventually it will stop, allowing for input. The most important thing to do is run the command "sudo apt-get update" which is really important because it will give the newest and most stable updates for your raspberry pi. "Startx" should be used to get into the desktop. Connect to the internet in whichever way is best, to make sure all is working properly.

PCB/Soldering

Next thing to start working on would be the PCB. The PCB provided to us by the Humber Prototype Lab is what we used to hold majority of our project. First download the program EAGLE, and the required board and schematic file which is provided by Humber College. The schematic and board file are just a generic student file so the name will need to be changed. Lastly we went to the prototype lab and asked what else they needed for us to print our board, and they asked for these two files and the rest was done by them!

When soldering the PCB, Vlad and Kelly at the Humber Prototype Lab really helped out. If you had a question on how to test it or where to solder some things, they would help you out. Majority of the soldering was done while looking at the reference model they had at the lab, we also used the solder and soldering iron they had there so we didn't need to buy our own. Once done, you should consult to Vlad and Kelly to make sure you've done it right, they will show you how to properly test it to make sure it works. Below is an image of the board file.
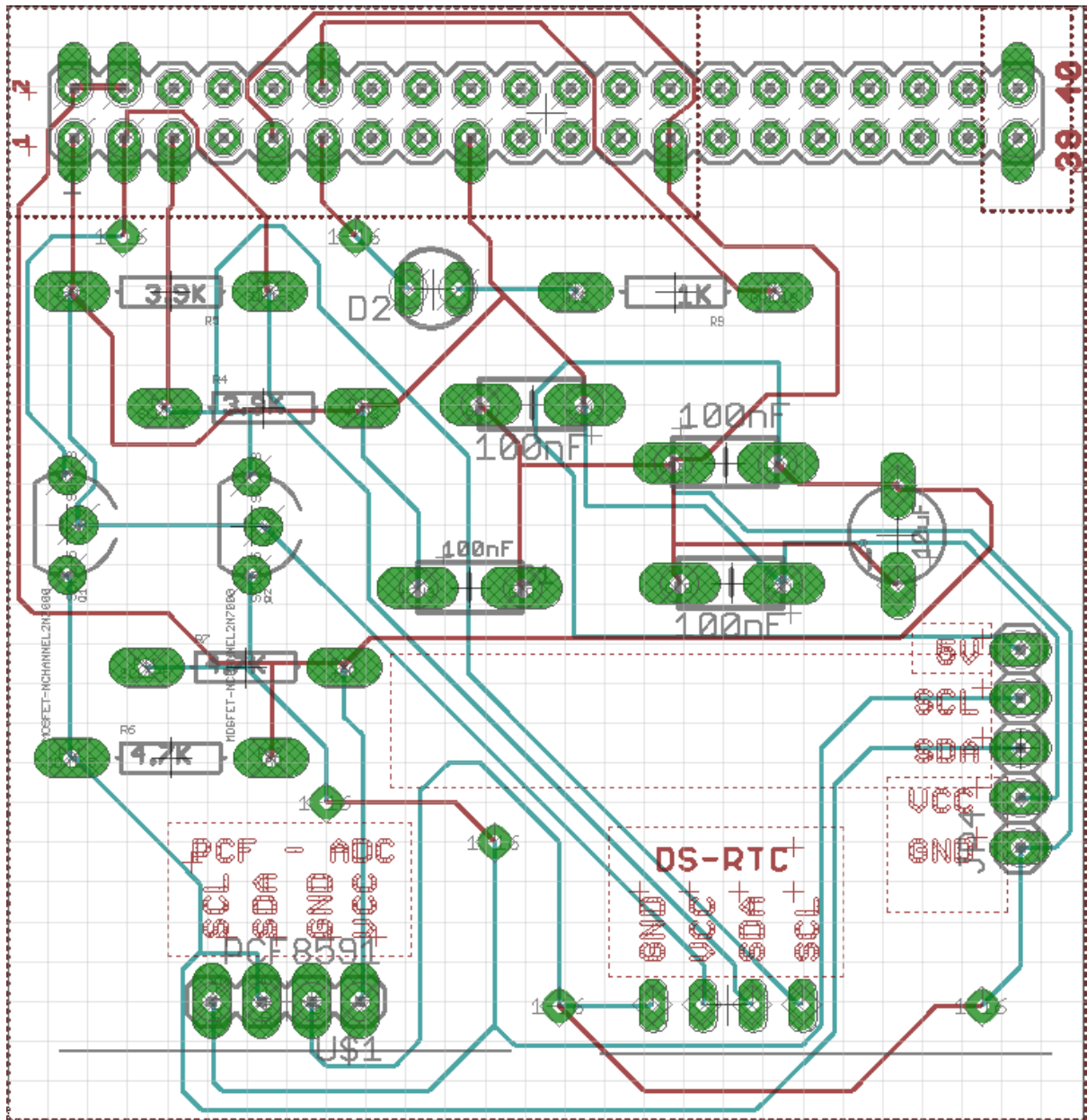
Figure 2: Schematic For Custom PCB

Assembling and Power Up

When the Raspberry Pi is setup, and the PCB is finished and soldered you can now set up the rest of the project. Get the Modular Sense Hat and connect it properly into the header labelled PFC-ADC on the board. Take the MQ-3 sensor, connnect the positive to 5V, negative to ground and analog signal pin to the AIN1 pin on the Modular Sense Hat. Take the Heartbeat sensor, connect positive to VCC in the header labelled DS-RTC on the board, negative to ground, and signal ping to the AIN2 pin on the Modular Sense Hat. Lastly, using the 24-pin GPIO header on the PCB board, connect it to the Raspberry Pi firmly so it's set.

Once everything prior was completed, the only thing left is power up. Do a quick check and make sure everything is wired correctly, and make sure that there are no problems. Go ahead and plug the USB's into the Raspberry Pi, power it on and see if it works. You can see if the sensors are being detected by using the command "i2cdetect -y 1" on a terminal and you should see 48, which corresponds to the Modular Sense Hat! Below is the code we created and can be ran by putting it in a file named lifelines.py, and can be ran by typing "python lifelines.py" in terminal.

```
import time

import RPi.GPIO as GPIO

GPIO.VERSION

GPIO.setmode(GPIO.BOARD)

GPIO.setup(11,GPIO.OUT)

GPIO.setup(12,GPIO.OUT)


from smbus import SMBus


bus = SMBus(1)


#read the sensors given the pin

def read_ain(i):

    global bus
```

```
        bus.write_byte(0x48, i)

        bus.read_byte(0x48)

        bus.read_byte(0x48)

        return bus.read_byte(0x48)



    while(True):

        alcohol = read_ain(2)*0.001      #read the alcohol sensor on pin 2

        heartrate = read_ain(1)          #read the heartrate sensor on pin 1


        print "------------------------\n" #print alcohol reading

        print("Alcohol Sensor: {0:.3f}%".format(alcohol))


        #turn on the LED when alcohol level is above 0.08%

        if(alcohol>0.08):

        GPIO.output(11,0)

        GPIO.output(12,1)

        else:

        GPIO.output(11,1)

        GPIO.output(12,0)


        #print heartrate reading

        print("Heart Rate Sensor: {0:.0f} BPM\n".format(heartrate))

        time.sleep(1)#update every 1 second
```

Connecting To The Database

At first, we could not decide on how to connect the mobile application and the hardware to a database. We were originally going to use an SQL database that was going to be ran off a server, but then we found out about Firebase. Firebase is a mobile and web application platform which includes a Realtime Database, which uses JSON instead of SQL to store data, and Firebase Auth, a service that is able to

authenticate users locally to access specific data from the database. We decided to go with Firebase because it is free and we do not have to rent and set up a server.

Connecting Firebase to the android application was easy because Firebase is a built in option in Android Studio. We had to set up the database rules so that the database read and writes from the correct user profile that is currently logged in to the app. There is an assistant in Android Studio that helps you with setting up Firebase and adding all of the correct dependencies and libraries in the build.gradle file. We followed the tutorial from the Firebase website to get authentication with email and password and on adding data to the database.

Connecting the hardware to the database was a little bit more difficult. We used a python wrapper called Pyrebase that uses the REST API to access the data from the Firebase database. We had to install Pyrebase to the Raspberry Pi and then import and add the configuration for Firebase that you can find in the Firebase console into the python script. When it detects that a test finishes, it pushes the data to the database and then the mobile application can read and display it in the app.

2.5.5 Testing

Unit Testing

For unit testing there wasn't a lot of things to be done. The unit testing we did was the individual sensors. To test the MQ-3 sensor, we created a small and short code which allows us to see a continuous loop for the MQ-3 reading, which allowed us to see the readings change if you blow into it. Considering you're not intoxicated, the readings shouldn't go up while the code is running. You can now take a rubbing alcohol bottle, or to be safer you can just use a hand sanitizer bottle and squeeze out some air into the sensor, then the reading should start to go up. For the heartbeat sensor all we needed to do was alter our previous code for the MQ-3 sensor, run it and now put a finger onto the green light of the sensor and watch the screen. If done correctly the reading should drop down to a number lower than 10, and then give you a more accurate reading of something between 70-80 after five or so seconds. If both of these work properly, well done!

Production Testing

Now that we're at the last step and we're sure that everything is working properly, all we had to do was combine the two codes from unit testing and run it! If everything works properly the code should work and you should be getting successful readings from both sensors if working properly. we also added a code for the LED on the PCB board so now, the LED will be red if a impossible reading for the heart beat is detected, so if the reading is less than 60 or greater than 100, the reading is inaccurate or your heart rate is not stable.

To end this project off you can create your own case using the program CorelDraw, which allows you to create a sketch for your case. The own personal touch is nice, and you can add just about anything. Once done you can save the file you created on CorelDraw into a pdf, and bring it to the prototype lab and see if everything is correct for it to print. It will just take a couple of minutes, then you'll have everything done and in your hands. All you have to do is put the Lifelines: Breathalyzer into the case, and make sure everything is where you want it! That's it!

2.6 Schedule & Progress Reports

Phase 1

- Project Selection & Individual Proposal Due

  Tue. 9/6/16 - Tue. 9/13/16

- Individual Project Schedule

  Tue. 9/16/16 - Tue. 9/20/16

- Status Meeting

  Tue. 9/20/20

- Individual Budget

  Tue. 9/14/16 - Tue. 9/27/16

- Individual Progress Report (Components Acquired Milestone)

  Wed. 9/28/16 - Tue. 10/11/16

- Status Meeting II

  Tue. 10/4/16

- Individual Progress Report II (Mechanical Assembly Milestone)

  Wed. 10/5/16 - Tue. 10/11/16

- Individual PCB due (Power Up Milestone)

  Wed. 10/12/16 - Tue. 10/18/16

- Status Meeting III

  Tue. 10/18/16

- Group Placard

  Thu. 10/20/16 - Tue. 10/25/16

- Individual Progress Report III

  Sat 10/29/16 - Tue. 11/01/16

- Hardware Demonstration (November 12th, 2016)

- Peer grading of demonstration

  Thu. 10/27/16 - Tue. 11/8/16

- Individual Build Video

  Wed. 10/12/16 - Tue. 11/15/16

- Status Meeting IV

  Tue. 11/15/16

- Individual Progress Report IV

  Wed. 11/16/16 - Tue. 11/22/16

- Individual Presentations, Individual PowerPoint

  Thu. 11/10/16 - Tue. 12/06/16

- Individual Video Script, Final Filming and Demonstration

  Mon. 11/21/16 - Tue. 12/13/16

Phase 2

- Scheduling and Group Meetings

  Mon. 1/09/17

- Group Project Status Update

  Fri. 1/13/17 - Mon. 1/16/17

- App, Web, and Database SRS

  Mon. 1/09/17 - Mon. 1/23/17

- Group Project Status Update II

  Fri. 1/27/17 - Mon. 1/30/17

- Group Project Status Update III

  Thu. 2/2/17 - Mon. 2/6/17

- App, Web, and Database Independent Demonstration

  Tue. 2/14/17

- Group Poject Status Update IV

  Sat. 2/25/17 - Mon 2/27/17

- Group Integration

  Tue. 2/14/17 - Fri. 2/17/17

- Group Project Status Update V

  Mon. 3/6/17 - Mon. 3/13/17

- Group Troubleshooting

  Wed. 3/1/17 - Mon. 3/20/17

- Group Project Status Update VI

  Fri. 3/24/17 - Mon. 3/27/17

- Project Demonstration at Open House (April 8th, 2017)

- Group Presentations

  Mon. 4/10/17

- Group Final Report

  Sat. 3/18/17 - Mon. 4/17/17

- Group Video Script, Final Filming and Demonstration

  Sat. 3/18/17 - Mon. 4/24/17

As seen in the project schedule above, everything is pretty layed out. Planning ahead was crucial to get this project organized and properly done. The project would not have been done as well as it did if we did not follow this schedule as much as possible. Having all these days written down so we can follow them allowed us as a group see what needs to be done, and when it needs to be done by so we are able to see when we have extra time to work on something that will take a lot more work. For example, in phase one We had set our schedule so that we start our build video a month before it was actually due.

Knowing that this task would take a long time we allocated more time to it, and was able to work on other tasks while working on the build log slowly and not all at once.

Although the schedule might not have been followed to the exact dates, we believe that the main benefit from this schedule is for time management and making sure you're able to follow tasks and schedules set. Even though you might not be able to do things exactly and always be on time, it is more beneficial if you're more disciplined enough to fall behind, but make up for it and catch up when you get knocked down. This schedule that we had to make also helps us in the long term and towards the end where we can see how long it took us to do some tasks and see how we could improve our time management in the past. A schedule was also made in Software Project class where we had to make the mobile application. Again, this helped us prioritize our application when we needed to, but also work on labs and assignments from that class at the same time.

Progress Report [11/10/2016] - Sent by Eugene Oliver

As of today there has been several changes to the initial plan of the hardware idea for Lifelines: the Portable Breathalyzer. After meeting with Kristian Medri about our proposal, we realized that a few things needed to be added and changed. Instead of going with just one sensor that looks at your alcohol level, my group decided to also add a heart rate sensor that will monitor your heart beat. We also realized that we needed to include a collaboration. We now are working in collaboration with the HRT (Hospitality, Recreation and Tourism) department here at Humber college. We will be looking at all the sensors they have at their center and we will also be able to ask them any questions that we have uncertainties with.

In terms of meeting the objectives based on the approval of our project proposal I am in pretty good standings. All the materials and parts needed for the project have either arrived in my possession or are currently on their way. To avoid any time delays and full lab problems, I came to school during our reading break (Thursday Oct 6th) to work on finishing the PCB and solder all the things together.

One problem that came up is with one our parts that are coming in. Our 7-segment display that we planned to include to display the readings will be coming in a month, which will impact the budget building time. To fix this Kristian Medri suggested to talk to Vlad and Kelly to see what our other

options are while waiting for these parts. In terms of finances, we were able to find cheap resources to include in our project since we will not be receiving any project funds. Either than the actual Raspberry Pi kit that we had to purchase and parts kit we bought in first year, everything else was relatively cheap. My total budget including everything I bought in the past came up to around $360, but removing the things I already had, I just needed to spend a total of $63.69.

In terms of moving forward, my group will be able to commence our project building plan. When looking at my designed GANTT chart, I can see that if I am able to follow my schedule I will be able to complete it on time. My next step in this project is to go about the procedure to test the PCB board and make sure nothing is wrong with it. Lastly, as time goes on I plan to continue documenting any work that has been done, package openings, and assembly that will eventually make up my build video.

Progress Report [08/11/2016] - Sent by Eugene Oliver

As of today there has been some updates and steps moving forward with the Lifelines: Breathalyzer hardware project. All parts that I believe are needed for the project have come in, so all building can be done and started. With the open house coming up this Saturday I plan to have a lot of work on my hands to make sure that everything I intend to show is working successfully.

In terms of meeting previous objectives and guidelines prior to today, everything has been working well and in time. Although I was given some issues when booting up and getting the PCB working with the sensors at first, I was able to consult Vlad and Kelly to see what the problem was when seeing the data on the screen. It turns out that one of my connections were shorted out because of a small strip of solder which connected two of my GPIO pins. This fixed the error of my sensors not being detected with the "i2cdetect –y 1" command. After fixing this problem the light and temperature sensors were giving proper readings and the LED blink test also worked as well.

The following weeks after the power up some projects were also due. I created the Placard for the breathalyzer which gives a quick and short description about the project and what we plan to do and use it for. It also highlights the main specifications and the names of the people who are working on this project with me. Lastly, I created a quick 30-second video which shows information on some of the things that were done in the previous weeks. It shows the parts that have came and arrived in the mail,

it also demonstrates a quick light test, then ends off with proof that the sensors are booting up on the PCB while connected to the Raspberry Pi.

In terms of moving forward, as stated before, I plan to get as much ready for the open house coming up. I've gotten data from my sensors but only on an analog reading which just shows 0's and 1's, my next step would be to get those values to actually show full numeric values so it's more useful. Lastly, as time goes on I need to continue preparing for my individual build instructions and final video filming in the last few weeks.

Progress Report [15/11/2016] - Sent by Eugene Oliver

As of today there has there has been signs of progression for the Lifelines: Breathalyzer. This past Saturday was the Humber Open House where we presented our projects so far and got feedback and some reactions from families and possible future Humber students. I also got to see other projects of my classmates and see where they are currently at with their projects as well. I received helpful feedback from other groups and Kristian that I will talk about in a future report. A lot of guests that came to the open house were happy to see that the project was working and that the sensors were cool to interact with.

So far, everything has been going smoothly. At open house I was able to demonstrate what I planned to show, and that was calibrating my heartbeat sensor and MQ-3 alcoholic gas sensor and getting successful readings from both of them. I also didn't need to bring a monitor and use the HDMI-to-VGA converter because I was able to display my raspberry pi onto my laptop screen instead. This saves me a lot of hassle by not needing to pull wires out of a computer monitor in classes or at home.

One problem that I encountered this past week was on how I was going to demonstrate getting readings from my alcohol sensor, considering that I wouldn't need to use an intoxicated person. I solved this problem by just using a rubbing alcohol bottle, but this could've been further fixed by using just a hand sanitizer bottle. By using a hand sanitizer bottle instead, it wouldn't be a big problem if a spill were to happen. Another problem I encountered was when I was first trying to get readings from the sensors. As stated in my last report I was only getting 0's and 1's and that was fixed by using the ADC in one of the sensors that was given with our PCB kit from the earlier weeks.

In terms of moving forward, I plan to create an acrylic case that will hold the project. I also plan to further calibrate the sensors to get more accurate readings, and try to get a graph that displays the data rather than just numbers. All parts have been received so I will not need to put anymore money into the project which ends the finances at around $63.

Progress Report [22/11/2016] - Sent by Eugene Oliver

There has been signs of progression in my project Lifelines: Breathalyzer. In the next coming weeks things will need to be done in order for it to be finished by week 15. Next week will be my presentation, so I need to make sure everything that needs to be done next week will be done.

Everything is still going smoothly and according to schedule. Last week I was starting to think about my case that I learned is mandatory for the presentation. I also need to get the LED working on the PCB board and somehow use it to incorporate it with the rest of my project. My plan right now is to get the LED working, corresponding to what the sensors are reading. So for example, I will make the LED be red if one of the values is too low, and green if it is a proper reading.

One problem that I encountered this past week was trying to figure out how to make a case. After drawing out my ideas on a paper I was then stumped on what to use to actually print my design. I figured out how to fix this problem by asking classmates who already have some of their case printed out, and I confirmed this was right by going to the prototype lab and asking Vlad/Kelly. Another problem I had was getting the LED on the PCB to blink using python. Before, we used the blink test using the code given to us, using C. I was able to get some sample code working by researching online and working with some friends last week and making sure it works. Now i just need to incorporate it into my own code.

In terms of moving forward, I plan to just stay on track to have everything done by the final week of school. As time is closing in on me, I'm starting to look back and see how well I followed my schedule and I'm proud of how far my project has come along. As stated in my last progress report, there were no additional finances to be added, so there are no changes for that.

Progress Report [30/01/2017] - Sent by Adriene Almacen

As of today we are currently on track with our project. Our project proposal with inline citation has been submitted as well as both the skeleton, and our software requirement specification. In the near future we plan to discuss our project with the HRT – Hospitality, Recreation and Tourism department at Humber College for guidance, help and additional planning.

In terms of our actual project, everything is on schedule. The layout for our Database and mobile application has been created and we just need to connect the following to our Hardware. We plan on having the data from the alcohol gas sensor and heart rate sensor display on our mobile application and store it on a database.

A problem that we have encountered and realized in the past weeks is that since we are using Eugene's hardware, the size of the acrylic case he made has his name on it from last semester. It is also a bit large in size, and in order to solve this we plan on creating a smaller more compact case. We also initially planned to have out Brethalyzer as a portable device for users to move and bring around easily. However, we currently power our Raspberry Pi with a Micro-USB power adapter plugged into a wall outlet. To fix this issue, we are planning to include a micro-usb portable charger to our budger/project that can power the Lifelines: Breathalyzer.

In regards to the budget, a small addition will be included in the future as we will be ordering a portable battery so that we can power our Raspberry Pi and make the device portable. This portable battery will cost somewhere between $20-$30 for a power bank strong enough to power the Raspberry Pi.

In the future, we plan to keep working on the things required to get us to the end goal of our Lifelines: Breathalyzer, along with doing thing things to complete our technical report. We hope in the near future to be able to merge our database, mobile application and hardware into a complete working project.

Progress Report [14/02/2017] - Sent by Ryan Do

As of today, we are currently on track with our project. We have added more to the structure of the Technical Report as well as the Abstract, the Introduction, and the Declaration of Joint Authorship. We are still planning to meet up with Humber HRT – School of Hospitality, Recreation and Tourism in the coming weeks to discuss our project and see what kind of equipment they use.

During the past two weeks, Eugene has been working on a new design for the acrylic case to house our portable breathalyzer. He has also been tweaking with the sensitivity of the alcohol gas sensor to get as accurate as we can. A problem that Eugene has encountered is that the potentiometer on the alcohol gas sensor is very sensitive, and a small turn is able to make the reading change drastically. To fix this, he has put a small piece of cardboard or paper on top and taping to prevent it from turning.

Adriene has been working on changing some aspects of the mobile app with respect to the design, trying to make it more user friendly. He is in the process of changing the main screen to be a tabbed layout with the three main functions instead of having a different page for each of the different functions. A problem that he has encountered is how to display the past records tab when the user is logged in, and hide it when the user is not logged in. He was able to deal with it by detecting if a firebase user was authenticated and logged in and then setting the visibility of the tab to zero.

I have been working on the JSON database from Firebase and adding the date and time of when the test was taken to be able to categorize and keep track of when the data was entered into the database. A problem that I have encountered is that since our device is not yet connected with our mobile app, I could not get any real data to be inserted to the database to see if the time and date function works. I have resorted to having a button add a set number to the database to check for the time and date functionality.

Altogether, we have been searching for an external battery pack and have narrowed down our choices to a select few batteries that we think would work well with our project. This will be the last item to add to our budget, which will bring our total to around $360.

In the coming weeks, we plan to continue to work towards integrating our portable breathalyzer with our mobile app along with working on the technical report. We hope that by the open house, we are able to fully integrate everything together and have most of the core components working.

Progress Report [07/03/2017] - Sent by Adriene Almacen

Dear Kristian,

As of today, we are currently on track in terms with our project schedule. We have recently merged a mixture of our Build instructions onto our Technical Report body. We are hoping to meet with a

representative of Humber HRT –School of Hospitality, Recreation and Tourism during our presentation to discuss our project and hopefully be able to collaborate with them.

In the past week, Eugene has worked on integrating the hardware into our new compact acrylic case by creating the proper dimensions for the case so that it is able to house the hardware as well as being a moderate size for it to be portable. In addition, Eugene plans on integrating an executable file that runs our breathalyzer program so that every time our breathalyzer turns on the script is always running.

Ryan has been working on integrating the hardware and mobile application together. He has done this by gathering the data from the Raspberry Pi, the data is then transmitted through wifi onto the database. Secondly, he is currently in progress of being able to transfer data from the database onto the mobile application using a JSON database from Firebase. The main data that is being stored onto the database is the user's info, Blood Alcohol Content (BAC) and Beats per minute (BPM) as well as the time and date of when the test was taken.

I have been working on the integrating a new layout for our application, with the feedback received from last semester. By changing the main screen to a tabbed style layout, it eliminates a majority of the buttons in our first design as well as having a screen with three main functions instead of having a page for each function. Furthermore, the login and past records function were moved to the toolbar for easy accessibility. The goal is to make our application more user friendly by having a simple, easy to read design.

In the next following weeks, we plan to continue working and finishing up our portable breathalyzer, mobile application and connecting it all to our database. Next week, we hope to finish a beta project with a majority of the components working to show at open house along with our meeting with a representative of Humber HRT.

Sincerely,

Adriene Almacen

Progress Report [21/03/2017] - Sent by Ryan Do

Dear Kristian,

As of today, we are currently on track with our project in terms of the project schedule. We have gone through and completed the OACETT basic requirements checklist while noting which areas we still need to improve on and complete. Last week, we had the opportunity to represent the Computer Engineering Technology program along with the Active House project for National Engineering Month. We only had a couple of people come in and talk to us about our projects, but they have provided us with their ideas and have asked good questions. We plan to have a tour of the labs of Humber HRT – School of Hospitality, Recreation and Tourism this week to get a glimpse of the equipment that are used to measure alcohol and heart rate in a lab.

During the course of the project, Eugene had some problems on the calibration of the alcohol gas sensor. The potentiometer that controls how sensitive the sensor is to gas particles, changes a lot when turned a little. We have decided to glue it so that it does not turn accidentally while working on the sensor. Another problem was getting the correct dimensions for the new acrylic case with holes for the standoffs for the Raspberry Pi. He went online to search for dimensions for the Raspberry Pi including the holes for the standoffs and measured it to fit the case.

In regards to the mobile application, Adriene had some problems getting the tab layout merged with our previous layout. Searching online, he found a guide showing how to incorporate a tab layout into the mobile application and is in the process of implementing it. Another problem Adriene came across was the permissions for accessing location and contacts. Moving from Android 5.1 (API 22) to Android 6.0 (API 23), google has changed the way that permissions are granted. In API 22, you would ask for permissions at the time when you install the app, but moving to API 23, you have to ask for permissions on the fly. Searching on the internet revealed the process for the permissions and it has been implemented into the app.

With the database and the python script, a problem that I had was how I was going to get the data from the Raspberry Pi and have it show up on the mobile application. I decided to use "Pyrebase", a python wrapper for the Firebase API, with my python script to push the data from the Raspberry Pi straight to the Firebase Database. The mobile application can then grab the data from the database and display it. Another problem I had was to detect when a test started and when it stopped. For the alcohol reading, I checked if the current value is higher than the previous value and set that as the max, and looped until the reading did not go any higher. For the heart rate, when it reads a very high value,

for example over 200BPM, the heart rate sensor starts reading values every second for 5 seconds and then the average set as the heart rate.

Financially, we have not added anything to the project since the addition of the portable battery pack and the new acrylic case. Our total for this project, which includes the Raspberry Pi, the sensors, the PCB, the external battery pack, and the parts toolbox, is around $360. In the coming weeks, we hope to be finishing the project and getting it ready for showcase at the Humber College Spring Open House on April 8th, 2017.

Sincerely,

Ryan Do

2.7 Software Project Reports

Fall 2016 Software Application Proposal for Lifelines: Breathalyzer

The following report is a proposal we had to submit to get permission to create the application that would be paired with our hardware project to create our systems project. This proposal had to include what problems our project were going to fix and how our application would help fix this problem. It also highlights the tasks each member of the group had to do, and we had to make sure everything was split as equally as possible.

Report: Software Proposal

Group Name: Designated Drivers

Project Name: Lifelines: A Breathalyzer Application

Team Members: Eugene Oliver (N01038196) Adriene Almacen (N01040646) Ryan Do (N01044391)

For our Fall 2016 Software project we are asking for approval to create an app that will allow users to find out their alcohol level and their heart rate. Depending on the reading the user gets, it will provide them functionality to do a certain amount of tasks. Our application will provide functionalities to call for help, either a friend or taxi and even see hotels in the area instead of putting themselves behind the wheel creating danger. This project will be closely related to the hardware project we are working on in CENG 317, which is a portable breathalyzer.

The mobile application will fix the problem of the large amount and frequency of drunk drivers that endangers their lives and the lives of others. Using this app, people can realize how much alcohol is actually in their system and from that, can create a solution for their safety instead of danger. Every year there is a call for help for people to stop drinking and driving, but yet every year there are accidents that always lead to a common problem of alcohol consumption behind the wheel. The largest age group for teens that drive drunk is 19 years of age, this age group is also a large target area for mobile application use. I believe that the more people who use this project will lower the rate of this problem and bring us one step closer to a solution for the real world problem of drinking and driving.

In terms of group member tasks, we plan to all work together and help each other in any way we can, whenever we can to achieve our final goal of a fully functional mobile app. Eugene will be mainly working with the UI and design. Adriene will also be looking at the design, along with working with application implementation to make sure there aren't any issues adding features together. Lastly, Ryan will be working with integration of external data sources along with working with issues and problems with accessing the remote server and database, he will also be scrum master.

Together as a group our main end goal is to create a complete mobile app throughout the next few months. Once completing both the hardware aspect along with the software/mobile app we plan to combine the two together to create a fully functional breathalyzer which will hopefully impact lives of users in a positive way. We hope our idea interests you and we look forward to communicating with you throughout the year.

Thanks,

Eugene, Ryan and Adriene

Distribution, Progress and Remaining Work After First Milestone

The following report is what we had to create together as a group and hand in when we reached the first milestone of the class. Since this was only the first milestone, not a lot of work was done on the application itself since we were still just trying it out and getting used to the functionalities since this was the first time we ever got experience with application creations. This report also explains what we plan on doing in terms of splitting up the tasks to make sure we are all on the same page on who is

doing what. Lastly, this report talks about what our goal is when finalizing the application and what still needed to be done at this point for the future.

Report: Distribution, Progress and Remaining work (First Milestone)

Group Name: Designated Drivers

Project Name: Lifelines: A Breathalyzer Application

Team Members: Eugene Oliver (N01038196) Adriene Almacen (N01040646) Ryan Do (N01044391)

For this milestone we managed to split up the work equally. Eugene created the GANTT chart which highlights the planned date ranges for the project completion. In this chart it also highlights important dates and meetings. Eugene also worked on the architecture design which makes it a little bit more clear how the app will work. Adriene worked on illustrating the design of how our application will look. He also highlights the tools that will be used in the project, what we will be implementing and what will be required when using the app. Lastly, our scrum master, Ryan worked on creating the minimum of two sets of image files for different resolutions of devices. He also worked on getting git access with android studio and successfully fetched and pushed code to the repo, he will also be demonstrating how to access it.

In terms of group member tasks, we plan to all work together and help each other in any way we can as discussed in our proposal. Eugene will be mainly working with the UI and design. Adriene will also be looking at the design, along with working with application implementation to make sure there aren't any issues adding features together. Lastly, Ryan will be working with integration of external data sources along with working with issues and problems with accessing the remote server and database, he will also be scrum master. As a whole we will constantly be debugging and troubleshooting, but meetings have been assigned to make sure we are always on the same page.

Together as a group our main end goal is to create a complete mobile app throughout the next few months. So far we have successfully thought of the idea, planned the design and implementation of features and are confident with our plans moving forward. After multiple planning stages we feel like we can start creating the app and push towards our beta release. If we are able to follow along with our project plan schedule and not run into many problems or issues, we feel like this will be a very successful application at the end when working with GPS, Bluetooth, Databases and other features along the way.

Designated Drivers,

Eugene, Ryan and Adriene

Individual Beta/Final Application Milestone Submission

The following reports are what we needed to hand in along with our application during the beta/final application milestone. In the first two reports, which were submitted by Eugene and Adriene at beta just highlighted what has been done in the time leading up to that day, what was done by each member, and what still needs to be done to the application. Ryan's report highlights what was done for the final application submission. At this point our application was done and Ryan's report is what he handed in along with our finalized application along with other technical documents.

Individual Beta Submission (Eugene Oliver)

Group Name: Designated Drivers

Project Name: Lifelines: A Breathalyzer Application

For this milestone we first came together as a group and talked about what would be best for each of us to do knowing that the best thing would be to split it up equally. We came to the conclusion that if someone were to do a hard part, they wouldn't need to do a lot in terms of quantity.

In terms of work progress as a group, we're in pretty good standings. We seemed to get most of the functionality of our app pretty much working. We had some bugs at the beginning but working as a team we were able to fix all the issues and get the app back on working track with no errors. We were able to get data from a database and connect to a server which allows us to register and login getting data for each user that registers on our app. We also changed the appearance of our application to make it look a lot nicer than before. Lastly we added functionality to our buttons which does unique things.

For myself I was assigned the task to get the parts of the app I was assigned to do this time was getting the google maps intent working when selecting finding hotels in the area. I also needed to get the runtime permissions working for CALL_PHONE for API 23+ to make sure that the permissions are granted if a phone call needs to be made. Lastly, I included some taxi's to call when the Call a Taxi button is

pressed. Along with the parts of the app, I also created the test plans which helped us as a group make sure that the application was running smoothly and doing what we wanted it to do.

My two group members, Adriene and Ryan did an amazing job with their contribution. Since we got majority of what we needed to do, I couldn't ask for a better job from the two of them. They did their part and I did my part which is why I feel like we were pretty successful as a group getting this done in such a stressful time during the school year. Also, we had good communication and we were able to assign the parts smoothly and help each other, so I feel like our group chemistry is definitely growing.

Thanks,

Eugene Oliver

Individual Beta Submission (Adriene Almacen)

Group Name: Designated Drivers

Project Name: Lifelines: A Breathalyzer Application

My Contribution for the app was to have the user enter in the users name and phone number and save it into the phones contact list.

In the near future I plan on adding my hardware project to connect with my app. Where I will be able to get appropriate readings and display them on my app and store it on the data base. As for my teammates, Ryan was the scrum master and was responsible for handing everything from work distribution to task management. He was also responsible for Connecting to a database through a login. He also worked with storing data in the database. Eugene was responsible for fixing the layouts, adding functionality for the taxi and hotel pages, and adding layouts for landscape and tablet landscape.

We managed to complete a working app in time for the milestone.

Thanks,

Adriene Almacen

Individual Final Submission (Ryan Do)

Group Name: Designated Drivers

Project Name: Lifelines: A Breathalyzer Application

For the final app, my contribution was to get the database running and having the signup/login pages working with the user able to use the Remember Me option. Originally, I was planning to use a local SQLite database for this but then I found out about Google's Firebase with their login and real-time database services. I have successfully registered users and have gotten their data go to the database and get their hardcoded past results from the database to show in an activity. My team members Eugene and Adriene did a great job in getting their assigned parts finished and pushed to their GitHub branch. We had great communication through chat and VoIP and we asked each other if we had any questions about the app or did not know something.

Thanks,

Ryan Do

3. Conclusions

This project of an affordable, portable breathalyzer has been created in hopes to lower the rate of drinking and driving. This breathalyzer can determine the users blood alcohol content (BAC) as well as their beats per minute (BPM), so that users can determine beforehand if they are able to drive home safely. When the alcohol or heart rate test is taken, data is retrieved from the sensors and stored on a database. In addition, the data is also displayed on an android application along with other information about when the test was taken. Furthermore, the android application will aid the user by allowing them to determine if it is safe to drive and will be given options such as calling a friend, calling a taxi or seeking the nearest hotels in the area. The final version of our project meets all the requirements that needed to be accomplished.

4. Recommendations

Currently, the hardware that we created can be better when talking about recommendations. If this project were to be reproduced multiple times and made perfect, a few changes and touch ups would not be a bad idea. A smaller more compact case can be created to make the project even smaller and portable when being looked at in final stages. That being said, instead of having a bunch of wires inside the case, you can create another circuit board which helps reduce the wiring and makes the project look a little more neat. This also allows an idea of neatness and the aspect of "plug-and-play". Although the sensors were pretty cheap, you would be able to order and buy the sensors in bulk so that you would be saving a fair amount of money by receiving a discount per sensor, and a cut in shipping cost. Also, since the program was created to fit the specific needs of our project, it can be further refined and customized to fit the end user's needs. For example, you can make the database not accept values that are too low, or too high and can tweak it to make sure it's only reading what you want it to read. The portable charger does not need to be the same and if you want to use a cheaper one you can. The only restriction is that it must output 5V/2.4A so that it provides enough power to the raspberry pi and is safe enough to keep it running. Lastly, more things can be added to make sure the parts inside the case do not move around a lot. This can be fixed by fastening the sensors with screws instead of tape. Once the project meets your standards then you can go about improving and making it even better.

5. References

Brüser, C., Kortelainen, J. M., Winter, S., Tenhunen, M., Pärkkä, J., & Leonhardt, S. (2015). Improvement of force-sensor-based heart rate estimation using multichannel data fusion. In IEEE Journal of Biomedical and Health Informatics (Vol. 19, pp. 227–235). https://doi.org/10.1109/JBHI.2014.2311582

Sakairi, M. (2012). Water-cluster-detecting breath sensor and applications in cars for detecting drunk or drowsy driving. IEEE Sensors Journal, 12(5), 1078–1083. https://doi.org/10.1109/JSEN.2011.2163816

Venugopal, M., Feuvrel, K. E., Mongin, D., Bambot, S., Faupel, M., Panangadan, A., … Pidva, R. (2008). Clinical evaluation of a novel interstitial fluid sensor system for remote continuous alcohol monitoring. In IEEE Sensors Journal (Vol. 8, pp. 71–80). https://doi.org/10.1109/JSEN.2007.912544