The Breathalyzer

From: Maheshwerie Samaroo, Karandeep Singh and Mohita Prabhakar  
Discipline: Computer Engineering Technology  
Date: March 26th, 2018

# 

# Declaration of Joint Authorship

The team “MKM” which consists of Maheshwerie Samaroo, Karandeep Singh and Mohita Prabhakar confirms that the project of the 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. Karandeep Singh is working with calibrating the sensors and was in charge of hardware design and helped with the app layouts. Maheshwerie Samaroo is in charge of mobile application design and maintenance and is handling the database. Mohita Prabhakar is working with the database in terms of setting it up, connecting it with the app and maintaining it. The distribution of testing the hardware and software for bugs and issues will be discussed by the three of us and worked on all together. Before any changes are made on the project, a group consensus has to be made. Links to articles/websites that are being used for guidance and help has been placed in the reference area of this report.

# Approved Proposal

5 February 2018

Proposal for the development of Breathalyzer

Prepared by Karandeep Singh, Maheshwerie Samaroo, Mohita Prabhakar  
Computer Engineering Technology Students  
https://github.com/N01150244/pulsesensor

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 the following sensors and actuators Heart Rate Educational Starter Kit, MQ3 Alcohol Sensor and Pulse Sensor. The database will store Readings from the heart rate/pulse sensors and MQ3 Alcohol Sensor. The mobile device functionality will include Allowing a user to sign-up or login to an existing account, view their current results as well as past results, call emergency contacts and call an UBER cab (if needed). and will be further detailed in the mobile application proposal. We will be collaborating with the following company/department Prototype lab, Humber Parts Crib and Humber Tech Group. In the winter semester we formed a group, who are also building similar hardware this term and working on the mobile application with group members Karandeep Singh, Mohita Prabhakar and Maheshwerie Samaroo. The hardware will be completed in CENG 317 Hardware

Production Techniques independently and the application was 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 2 or 3 student group.

Background

The problem solved by this project is We all want a solution to DUI (Driving under the Influence) or to at least reduce the number of deaths caused by it. Today, we see too many accidents being caused by DUI. Innocent lives are lost due to a drunk driver. Families are emotionally/financially stressed due to the loss of a loved one. We, the MKM Developers, intend to use our project, "The Breathalyzer", to help reduce these occurrences as well as give users a general idea on their BAC (Blood Alcohol Content) and pulse rate prior to them getting behind the wheel of a car.. A bit of background about this topic is The primary focus of our project is to address the issue of DUI (Driving under the Influence). Our target audience is the general public. The main reason for the development of our product, is due to the fact that many lives are being lost as a result of DUI (Driving under the Influence). With our product, we intend to combat this issue and reduce the statistics. We intend to reduce the number of deaths and prevent occurrences of DUI (Driving under the Influence). It is stated that the target audience is the general public. What this means is that, anyone can use this product. This product can be taken along with someone who decides to go to the Bar or to the Club or to any event that involves alcohol consumption. The project involves the integration of the mobile application which was built along with the hardware component. Basically the user will be required to blow into the alcohol sensor as well as use the provided pulse sensor. The sensors will capture the readings which will then be

pushed to a database. The mobile application will then pull the data from the database and display the readings via the application. Once the user is above the legal limit, he/she has the option to either call an Emergency Contact or request an UBER cab. The desired outcome of the overall project would be to assist in reducing DUI (Driving Under the Influence) occurrences..

Existing products on the market include [1]. I have searched for prior art via Humber’s IEEE subscription selecting “My Subscribed Content” [2] and have found and read [3] which provides insight into similar efforts.

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 me 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 16 Vrms 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 I have learned to potential employers.

The brief description below provides rough effort and non-labor 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.

Raspberry Pi 3 starter kit

XD-58C Sensor from Sparkykit heart rate sensor,MQ3 Alcohol Gas Sensor

Jumper Wires(Male-Female, Female-Female, Male-Male Jumper Wire Cables)

Heart Rate Educational Starter Kit

Concluding remarks

This proposal presents a plan for providing an IoT solution for This is an opportunity to integrate the knowledge and skills developed in our program to create a collaborative project which will lower the rate of alcohol consumption and bring us closer to a solution to prevent drinking and driving.

I request approval of this project.. 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 such as the initiative described by [3]. I request approval of this project.

References

[1] Heartbeats in Your Project, Lickety-Split ♥. (n.d.). Retrieved February 02, 2018, from https://pulsesensor.com/

#237238, M., #321089, M., #661774, M., O., #721982, M., #554862, M., . . . G. (n.d.). Pulse Sensor. Retrieved February 02, 2018, from https://www.sparkfun.com/products/11574

#637052, M., & O. (n.d.). Alcohol Gas Sensor - MQ-3. Retrieved February 02, 2018, from https://www.sparkfun.com/products/8880

Industries, A. (n.d.). Heart Rate Educational Starter Pack with Polar Wireless Sensors. Retrieved February 04, 2018, from https://www.adafruit.com/product/1077

[2] Institute of Electrical and Electronics Engineers. (2015, August 28). IEEE Xplore Digital Library [Online]. Available: https://ieeexplore.ieee.org/search/advsearch.jsp

[3] Wang, X., Jin, J., & Li, S. (2008, September 03). Measurement and analysis of heart signal based on the pressure sensor. Retrieved February 04, 2018, from http://ieeexplore.ieee.org/document/4618175/

Malathi, M., Sujitha, R., & Revathy, M. R. (2018, February 01). Alcohol detection and seat belt control system using Arduino. Retrieved February 04, 2018, from http://ieeexplore.ieee.org/document/8275841/

Kirtana, R. N., & Lokeswari, Y. V. (2017, June 08). An IoT based remote HRV monitoring system for hypertensive patients. Retrieved February 04, 2018, from http://ieeexplore.ieee.org/document/7944086/

# Abstract

As students in the Computer Engineering Technology program, we will be integrating the knowledge and skills which we have learned from our program into this Internet of Things themed capstone project. The internet connected hardware includes a custom PCB with the following sensors: Heart Rate Educational Starter Kit, MQ3 Alcohol Sensor and Pulse Sensor. The database stores readings from the heart rate/pulse sensors and MQ3 Alcohol Sensor. The mobile device functionality includes allowing a user to sign-up or login to an existing account, view their current results as well as past results, call emergency contacts and call an UBER cab (if needed) and was further detailed in the mobile application requirements section. We will be collaborating with the following company/department, School of Hospitality, Recreation and Tourism (HRT).

# Table of Contents

[Declaration of Joint Authorship 3](#_Toc509657971)

[Approved Proposal 5](#_Toc509657972)

[Abstract 17](#_Toc509657973)

[Table of Contents 19](#_Toc509657974)

[1. Introduction 21](#_Toc509657975)

[2. Project Description 23](#_Toc509657976)

[2.1 Product Perspective 23](#_Toc509657977)

[2.2 Product Functions 25](#_Toc509657978)

[2.3 User Classes and Characteristics 27](#_Toc509657979)

[2.4 Operating Environment 27](#_Toc509657980)

[2.5 Design and Implementation Constraints 27](#_Toc509657981)

[2.6 User Documentation 29](#_Toc509657982)

[2.7 Assumptions and Dependencies 29](#_Toc509657983)

[3. Mobile Application Requirements 31](#_Toc509657984)

[3.1 User Interfaces 31](#_Toc509657985)

[3.2 Software Interfaces 33](#_Toc509657986)

[3.3 Communications Interfaces 41](#_Toc509657987)

[4. Hardware Requirements 43](#_Toc509657988)

[5. Build Instructions 45](#_Toc509657989)

[5.1 Introduction 45](#_Toc509657990)

[5.2 Bill of Materials 47](#_Toc509657991)

[5.3 Time Commitment 51](#_Toc509657992)

[5.4 Mechanical Assembly 55](#_Toc509657993)

[5.4.1 Raspberry Pi Setup 55](#_Toc509657994)

[5.4.2 PCB Soldering 57](#_Toc509657995)

[5.4.3 Assembling the Project 59](#_Toc509657996)

[5.4.4 Power Up 63](#_Toc509657997)

[5.4.5 Connecting To The Database 71](#_Toc509657998)

[5.4.6 Unit Testing 75](#_Toc509657999)

[5.4.7 Production Testing 75](#_Toc509658000)

[5.4.8 Reproducible? 77](#_Toc509658001)

[6. Schedule & Progress Reports 79](#_Toc509658002)

[6.1 Fall Semester 2017 (Phase 1) 79](#_Toc509658003)

[7. Conclusion 87](#_Toc509658004)

[8. Recommendations 89](#_Toc509658005)

[9. Bibliography 91](#_Toc509658006)

[10. Appendices 95](#_Toc509658007)

# 1. Introduction

This report is a description of the processes followed for the development of a breathalyzer device. The targeted audience for this device will be the general public.

We all want a solution to DUI (Driving under the Influence) or to at least reduce the number of deaths caused by it. Today, we see too many accidents being caused by DUI. Innocent lives are lost due to drunk drivers. Families are emotionally/financially stressed due to the loss of a loved one. We, the MKM Developers, intend to use our project, "The Breathalyzer", to help reduce these occurrences as well as give users a general idea on their BAC (Blood Alcohol Content) and pulse rate prior to them getting behind the wheel of a car.

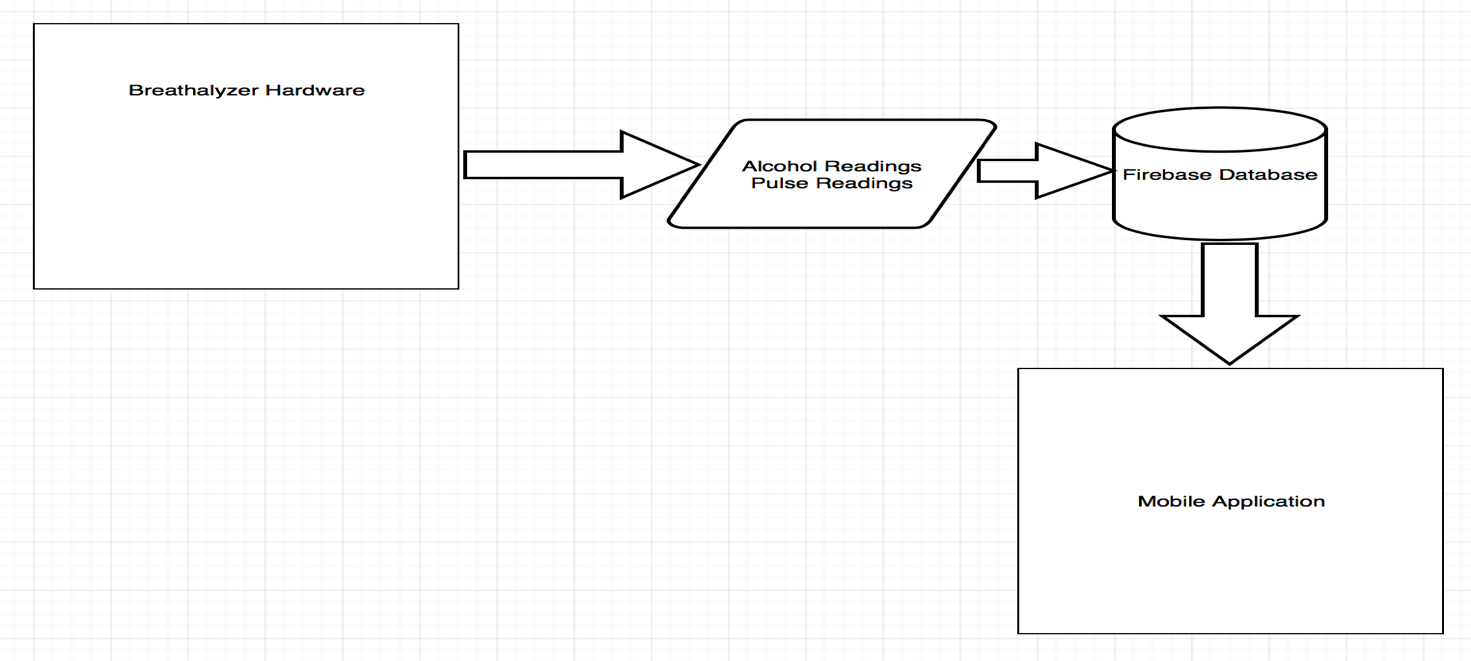
|  |
| --- |
|  |

# 2. Project Description

## 2.1 Product Perspective

This project encompasses the development of a new, self-contained product, “The Breathalyzer”. The product which is in development, requires software and hardware in order to perform its desired function. The software is in the form of a simple mobile application which is currently live on the Google Play Store as a simulator. The application serves as a display or medium through which the user can view the end result of using the entire product. For our purposes of explanation, we will refer to the Hardware as the “Larger System” and the Software as the “Smaller System”. The Larger System consists of three sensors which all collects/measures data which is then pushed to a Firebase Database. The function of the Smaller System is to display this data and give the user some additional options as well (as shown in the case below).

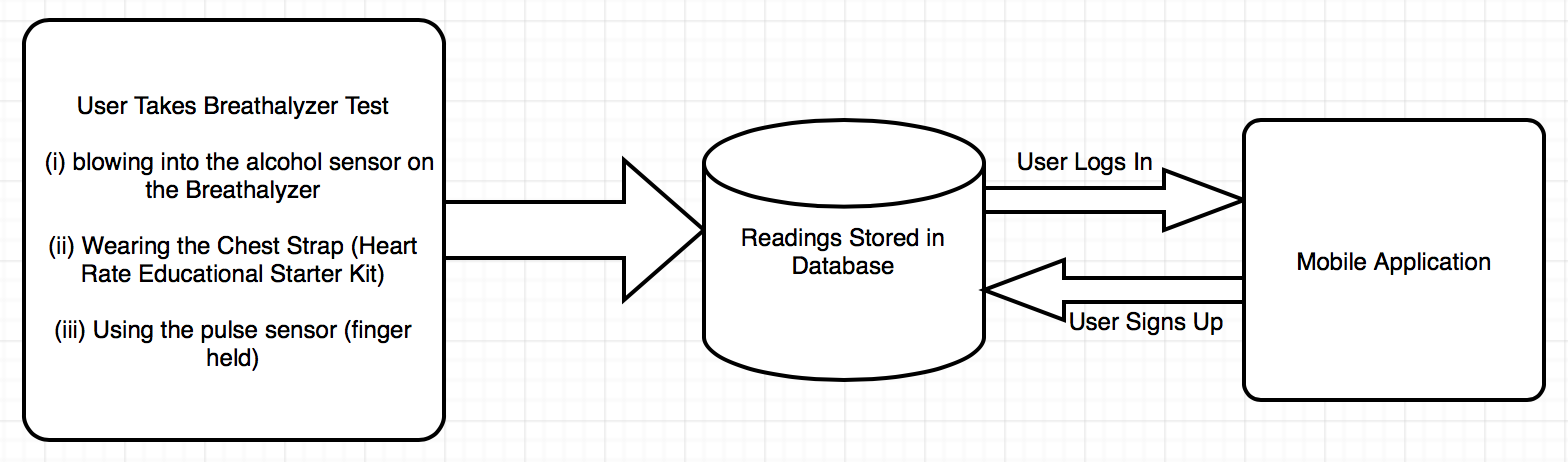
Case 1: John goes to the bar with some friends to have some drinks. It is now time to go home and John decides to take a Breathalyzer test to determine what his next decision should be. John takes the test by blowing into his personal Breathalyzer and the results show up on his mobile application. John is not safe to drive; he is over the legal limit. John now has the option to either call for an UBER cab or to Call two of his emergency contacts which he himself had stored previously. John calls a cab and gets home safely.



High Level Diagram Showing Data Flow

## 2.2 Product Functions

* Mobile Application – The User must login to his/her account (one time login unless logged out).
* Mobile Application – The user has a list of options each of which has a different purpose. The user can either see his/her results from the test, call a cab or call emergency contacts (which needs to be stored by the user).
* Hardware – The user must provide a breath sample (by blowing into the alcohol sensor of the breathalyzer) as well as taking two pulse tests.(one will be done using a chest strap and the other done by using a sensor which needs to be held by the user’s fingers)



## 2.3 User Classes and Characteristics

This product is meant to be used by the general public. Basically anyone from the age of 13 up is supposed to be able to use our mobile application and device. (The legal age for drinking alcohol in the province of Ontario is 19 years old and may vary from province to province. The age used in the statement above is just for reference purposes.)

## 2.4 Operating Environment

Raspberry Pi 3 with Raspbian along with Android mobile application, Firebase Database are the components of operating environment for the project.

## 2.5 Design and Implementation Constraints

With this project, there’s some implementation constraints. The mobile application for example, requires an active and steady internet connection in order for results to be pushed from the hardware to the database. An implementation constraint is the

accuracy of the sensors. The accuracy of the sensors depends on whether or not they have been properly calibrated.

## 2.6 User Documentation

Along with this software user will be equipped with the detailed build instructions and a technical report which will be available online: <https://github.com/N01150244/pulsesensor>

The user can also consult and get assistance from all of the online links mentioned in the references of the technical report.

## 2.7 Assumptions and Dependencies

Both the mobile application and the web application requires internet connection since the firebase database gets updated in real time. The python script will be used to store data in the database and can be retrieved on the application side via a php script.

# 3. Mobile Application Requirements

## 3.1 User Interfaces

This product comes with a mobile application with which the user interacts. Upon launching the application the user will be presented with a Login Screen with the option to create a user account if an account is not already existing. When the user logs in, he/she will be presented with a web view which shows the current news articles related to DUI (Driving Under the Influence). There is also a Navigation Drawer from which the User can choose one out of four options:

1. Start Test
2. View Past Results
3. Call a friend
4. Call Uber

## 3.2 Software Interfaces

Login Screen

The login screen is displayed if the user happens to select the login option from the home page. It is used by current account holders to login to their accounts. The username will be made to spec to allow an input of up to 60 characters while the password field will allow for up to 12 characters to be typed and will be disguised with asterisks to assist in security and prevention of hacking. Hints will also be used on each Edit Text so that users can differentiate between the Username and Password fields. The username and password will then be authenticated with entries which were stored in a firebase database.

Technical Note:

In order to disguise the password entries with asterisks, within the edit text declarations in the “activity\_login.xml” file, <android:INPUT\_TYPE=PASSWORD> will be used.

When the user clicks login, the following checks are performed:

1. If the user inputs an email address which either exceeds the 60 character quota or is in an incorrect format, (example doesn’t have the @ sign), then a toast will be displayed saying incorrect email has been entered and the user will have to re-enter the correct format.
2. If the user inputs an email address which isn’t associated with an account in the database, then a toast is displayed which says “The email address isn’t
3. associated with a user account. Please click on the register button to create an account”.
4. If an email address was provided but no password was entered, a toast will be displayed which says, “Please input your password”.
5. If the email address was provided, and it belongs to a registered account in our database but the password is incorrect, a toast will be displayed which says “Invalid Password entered. Please double check your entry”.
6. If the email address was provided and the password and they both match what was stored in the database, then the user is redirected to the menu screen.

Register Screen

The Register Screen will be launched if the Register option is selected from the Home Page. This screen will make use of various edit text fields to store the user’s information. The user’s information will be stored within a firebase database. The “Full Name” edit text will allow the user to input upto a maximum of 70 characters (Letters Only). The “Email” edit text will allow the user to input upto a maximum of 60 characters. The password fields will be validated to ensure that the user enters the same password twice. The Gender Field will be a dropdown list and DOB will make use of a calendar.

Technical Notes:

Limiting the Edit Text Fields to 60/70 chars will use the same format:

android:maxLength = “60” or android:maxLength=“70”

Limiting the Full Name field to letters only:

android:digits= “abcdefghijklmnopgrstuvwxyz”

Menu Screen

The Menu Screen is launched upon the user’s successful login or account creation. This screen makes use of a navigation drawer as well as a web-view. The web-view shows us news articles on DUI.

Start Test

This activity is started when the Start Test Option is selected from the menu. This activity has no functionality (until hardware is completed). For display purposes, random numbers are generated and stored into the firebase database.

Past Results

This activity displays the past results of a user.

Call a Friend

This activity is launched when the user selects the “Call a Friend” Option from the navigation drawer.

Add Contact

This activity is launched upon selecting the “Add Contact” option in the navigation drawer. This activity allows the user to input two emergency contact numbers and names. The input data is stored using shared preferences. After updating the contacts, the Call Friend activity is launched to see the changes that were made. The only visible changes will be the names of the buttons which will reflect the names of the emergency contacts stored in the “Add Contact” activity.

Call Uber

This option can be selected from the navigation drawer. Upon selecting, the Uber app will be launched if installed on your device. If the user doesn’t have the application installed, the play store will be launched.

## 3.3 Communications Interfaces

For communication interface, a database connection from python script is required to the database using various connect (), execute () and update function. On the application side, the application can be connected to the real time database easily through PHP Scripts. When the results are displayed from the python script after running the hardware, we can run a PHP query to extract the result we got from the database using REST API and update it into the application.

# 4. Hardware Requirements

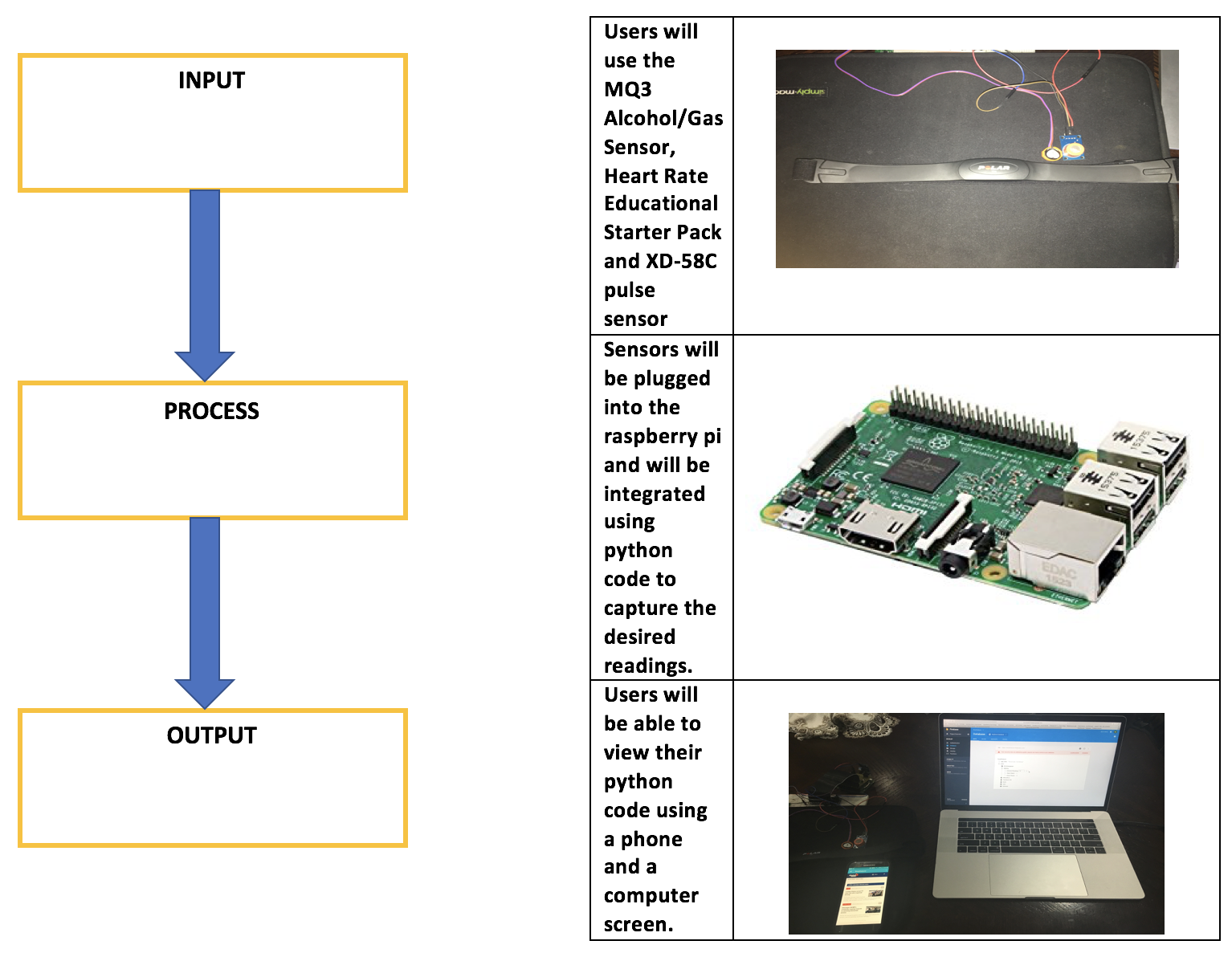
The Raspberry Pi software can be used through a LCD screen, a mouse and a keyboard. The screen is used to display sensor results as well. When results are transferred on to the application, these can be viewed via a mobile device using Android 5.0 or higher. This project makes use of three sensors. The first sensor is the MQ3 Alcohol Sensor. Currently, this sensor is being worked on by Mohita Prabhakar. There is a python script with which this sensor works. Basically a user can blow into the MQ3 sensor and a reading will be displayed and pushed to a Database. The second sensor is the XD-58C pulse sensor. This sensor is currently being worked on by Karandeep Singh. The user should be able to attach the sensor via a velcro strap around their finger and his/her pulse rate will be captured and pushed to a Firebase Database. The third sensor is the Polar Heart Rate Sensor which is commonly known as the “Heart Rate Educational Starter Pack”. This sensor is currently being worked on by Maheshwerie Samaroo. In this project, there are two pulse sensors which improves accuracy of the readings. One sensor may be more accurate than the other.

# 5. Build Instructions

## 5.1 Introduction

This page is created to help you to create your own Heart rate sensor project. In order to fix the problem of drinking and driving, we chose the Heart Rate Educational Starter Kit, Alcohol Gas sensor and Pulse sensor. Instead of people assuming that they are okay to drive with alcohol in their body, they can use our project to see if there is a large amount of alcohol in their body along with seeing their heart rate to determine whether or not they are legally able to drive. It's a cheap alternative, and isn't too hard to make yourself.

Using a gas and two heartbeat sensors, users can plug those into a PCB and use a raspberry pi to display the readings on either a computer or phone on our mobile application.



## 

## 5.2 Bill of Materials

Given below are parts and material required for this project. Most of these parts are pretty cheap which makes this project not too expensive, but that is because we already had our respective raspberry Pis, electronic parts kit, and PCB kits which were paid for as a part of our tuition.

|  |  |  |  |
| --- | --- | --- | --- |
| Item | Quantity | Cost | Supplier & Part Number |
| Raspberry Pi 3 Starter Kit | 1 | $79.99 | (Canakit) Amazon - B01CCF9BYG |
| XD-58C Sensor | 1 | $18.99 | (JutaTech) Amazon.ca - B01AUVMFIS |
| MQ3 Alcohol/Gas Sensor | 1 | $8.80 | (Creatron) creatroninc.com - SPEDE-000030 |
| Heart Rate Educational Starter Pack | 1 | $65.00 | (AdaFruit) adafruit.com - 1077 |
| Electronics Parts Kit | 1 | $119.99 | Humber - SKU #163 |
| Jumper Wires (120 pack) | 1 | $19.99 | (Elegoo) Amazon.ca - B01EV70C78 |
| 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 $360.

## 5.3 Time Commitment

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.

|  |  |
| --- | --- |
| ThingS you need to do | Time Taken To Complete (Approx.) |
| Looking for and Purchasing Parts + Delivery | 1.5 hours + 1 week |
| Assembling case and setting up Raspberry Pi | 1 hour |
| Editing your custom PCB | 30 minutes |
| Soldering/Testing/Troubleshooting your PCB | 3.5 hours |
| Creating a case for your project | 1 hour |
| Testing/calibrating the sensors | 2 hours |
| Setting up the project | 10 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.

## 5.4 Mechanical Assembly

To keep things simple, for mechanical assembly, everything will be broken into sections. First, there will be a description on how to setup your raspberry pi, what is needed to be done in order for it to work properly. Next, steps to create your own PCB will be detailed. After that, steps to connect and power up will be detailed.

### 5.4.1 Raspberry Pi Setup

Once you’ve acquired your Raspberry Pi, the first thing to do would be to get a keyboard and a mouse connected. Please note that whenever connecting any new device or making any new connections to the Raspberry Pi, please ensure that you Pi is powered off completely. For the Pi that was used for this project, what was noticed, was that when connections were made while the pi was powered on, the pi would freeze and become unresponsive. The next thing to do would be to insert the SD card which comes with the Pi. There’s many ways in which to connect the Pi to a display but in this case, a HDMI cable was used to connect to a monitor. The final thing would be to plug the Pi in. A black screen will be displayed at first, which is fine and absolutely normal. You will need to run an update which can be done by issuing the “sudo apt-get update” command. Please note that your Raspberry Pi must have an active internet connection in order for this to work. You can purchase a Wi-Fi dongle separately for this or use a LAN cable. In this case, we had neither so using one of our mobile phones, we used a feature called “Mobile Tethering” which enables the phone to share its internet connection when it is connected via USB. The final step would be to download the python code needed for this project which can be found on the GitHub. (https://github.com/N01150244/pulsesensor/blob/master/Breathalyzer.py)

### 

### 5.4.2 PCB Soldering

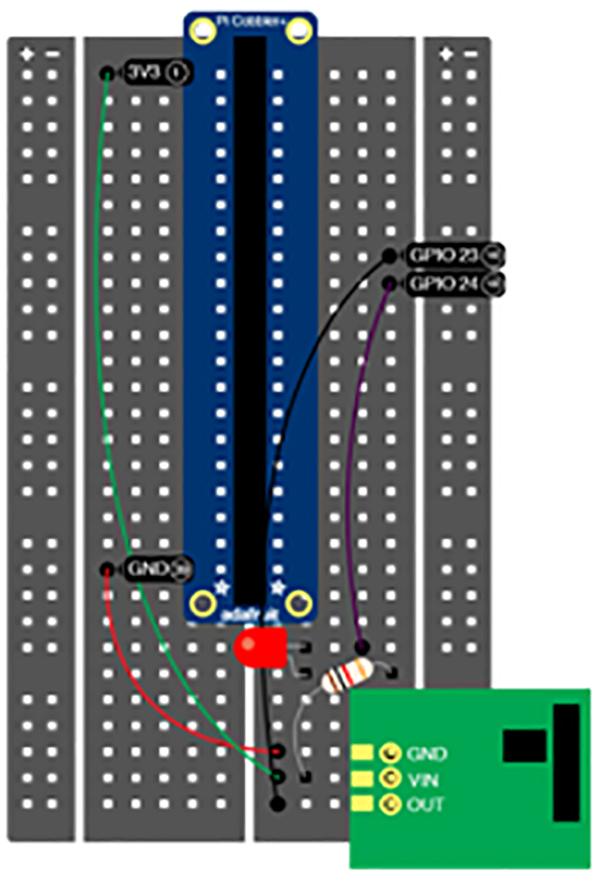
The next thing to start working on would be the PCB (Printed Circuit Board). The PCB which was provided by the Humber Prototype Lab is what was used to facilitate majority of the project. First you need to download the software EAGLE and then open the appropriate files. The schematic and board files were provided to us by our professor.

Assistance was given to us when needed by Kelly and Vlad who both work at the Prototype Lab. They were more than willing to assist whenever students needed it. Most of the soldering was done by looking at a reference model. Tools were provided by the lab, free of cost for the soldering of the PCB. Here’s what your PCB should look like when you’ve finished.

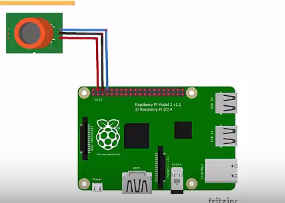


### 5.4.3 Assembling the Project

Once the Raspberry Pi is properly set-up, and your PCB is finished soldering, the rest of the project can now be assembled. The PCB needs to be “plugged in” directly to the raspberry pi. In this case, we plugged the PCB into the Raspberry Pi and then connected a pi cobbler plus breakout cable which we then connected to a breadboard. This makes it easier for you to make your connections. Note that it isn’t a requirement for you to buy the pi cobbler and breakout cable. The PCB must be connected to the header which is labelled “PFC-ADC”. Please remember that the Raspberry Pi must be completely powered off. The first sensor which we will speak about will be the XD-58C sensor (pulse rate sensor). The XD-58C sensor has three cables, which are colored purple, red and black. The red and black cables must be connected to 3.3V and GND respectively using the DTC header on the PCB. The purple cable must be connected to the AIN1 pin on the PCB. Please make sure that the Raspberry Pi is powered off completely before making these connections or else you will be at a risk of shorting a connection on the board. The second set of sensors which were used were the Polar T34 Heart Rate Transmitter and the Polar Heart Rate Receiver. Connecting the Heart Rate Receiver is simple. Below is a diagram which was used.



The third and final sensor which was used was the MQ3 Alcohol/Gas Sensor. The Raspberry pi is connected with the Mq3 sensor in such a way that VCC, D out and A out of the MQ3 Sensor is connected to the pin Vcc (4),Ground(6) and GPIO14(8) respectively as given below in the picture.



### 5.4.4 Power Up

Once all the connections have been made, the last thing left to do would be to plug the Raspberry Pi in. A quick check can be done to ensure that the sensors are in proper working order. The command “i2cdetect -y 1” can be issued in the terminal and 48 will be returned. The next step will be to run the code previously downloaded from the GitHub. First you must navigate to the location of the code via the Terminal and then issue the command “python Breathalyzer.py”. If the sensors are correctly connected, then the code should work perfectly.  
Below is a copy of the code.

from firebase import firebase

firebase = firebase.FirebaseApplication('https://breathalyzer.firebaseio.com/')

import time

import RPi.GPIO as GPIO

GPIO.VERSION

GPIO.setmode(GPIO.BOARD)

GPIO.setup(11,GPIO.OUT)

GPIO.setup(12,GPIO.OUT)

GPIO.setup(8,GPIO.IN)

GPIO.setwarnings(False)

from smbus import SMBus

bus = SMBus(1)

def read\_ain(i):

global bus

#bus.write\_byte\_data(0x48, 0x40 | ((i) & 0x03), 0)

bus.write\_byte(0x48, i)

bus.read\_byte(0x48)#first 2 are last state, and last state repeated.

bus.read\_byte(0x48)

return bus.read\_byte(0x48)

applicant = raw\_input("Enter your username:")

#while(1):

#heartrate = 101

#heartrate = read\_ain(1)

input = GPIO.input(8)

if input == 0:

print "Alcohol is detected"

print (input)

result2 = firebase.patch('https://breathalyzer.firebaseio.com/user/'+applicant,{'Alcohol Reading':input})

else:

print "Testing -----------"

print ("Alcohol not Detected")

result2 = firebase.patch('https://breathalyzer.firebaseio.com/user/'+applicant,{'Alcohol Reading':input})

#j+=1

#time.sleep(2)

i = 0

while(i<3):

heartrate = read\_ain(1)

print "Testing HeartRate-------------\n"

if(heartrate<60) or (heartrate>100):

GPIO.output(11,0)

GPIO.output(12,1)

else:

GPIO.output(11,1)

GPIO.output(12,0)

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

i+=1

time.sleep(2)

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

result = firebase.patch('https://breathalyzer.firebaseio.com/user/'+applicant,{'Heart Rate1':heartrate})

|  |
| --- |
|  |

### 5.4.5 Connecting To The Database

Originally, when starting the project, our biggest question was how to push data from our hardware to the database and pull that in our web page and mobile application. We were thinking of using Shared Preferences for our mobile application or SQL. We had heard about Firebase but never thought about using it. We took the opinion of one of our professors from our Software Project class and he suggested that we use Firebase since it is relatively free and cheap to use.   
  
Luckily for us, Firebase has built in functionalities in Android Studio. Firebase works with certain rules for reading and writing data from and to the database which we had to specify.

Connecting firebase to our hardware wasn’t that difficult as well. We made use of Firebase’s REST API to make our connection. These are the steps to getting your raspberry pi setup for interaction with Firebase:

1. Install python-firebase using pip:

pip install -e git://github.com/mikexstudios/python-firebase.git#egg=python-firebase

1. Then simply import firebase at the top of your python script:

from firebase import Firebase

1. and then instantiate Firebase, passing in your root url:

f = Firebase('https://SampleChat.firebaseIO-demo.com/')

1. Now call the different methods of the Firebase class (see the Firebase REST API page: <http://www.firebase.com/docs/rest-api.html> and the source of firebase/\_\_init\_\_.py for what methods are available and how to call them). For example, to push a list of data:

f = Firebase('https://SampleChat.firebaseIO-demo.com/message\_list')

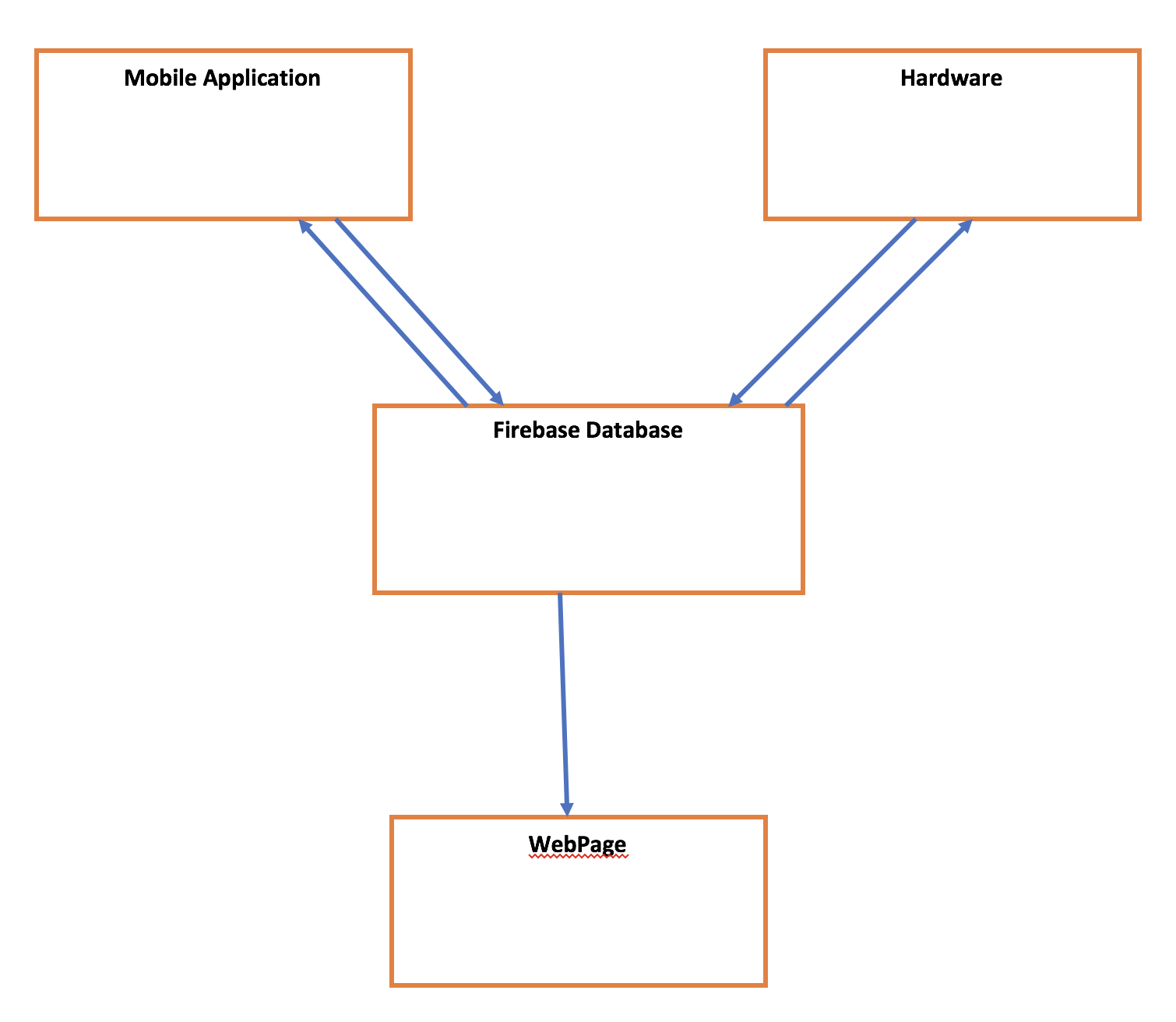
r = f.push({'user\_id': 'wilma', 'text': 'Hello'})

### 5.4.6 Unit Testing

For unit testing, we individually tested each sensor with individual pieces of code before combining into one piece of code. For the XD-58C sensor, the sensor should detect a heartbeat and it’s reading should be shown on the terminal screen. The same conditions are applicable to the other two sensors.

### 5.4.7 Production Testing

This is the final step. At this stage everything should be in perfect working order. We’ve tested our code and sensors based on our Firebase Database. Basically if everything is in working order, then results from the sensors should be posted to our real time Firebase database. Also, this project works hand in hand with a mobile application which can be downloaded from the Google Play Store. Below is a simple diagram for you to understand the process flow of our project. The Webpage is a current work in progress and will be updated on our GitHub.



### 5.4.8 Reproducible?

If this guide is followed step by step, then you shouldn’t have any issues with the set-up. As long as you have your parts and components, this project can be completed within a week. The best advice especially if you’re trying to meet a deadline, is to get started as soon as you receive your parts and be sure to track and slot your time wisely.

# 

# 6. Schedule & Progress Reports

## 6.1 Fall Semester 2017 (Phase 1)

* Project Selection and Project Proposal Due

Start : Monday 9/11/17

End: Sunday 9/17/17  
Due: Monday 9/18/17

* Project Schedule

Start: Monday 9/18/17

End: Sunday 9/24/17

Due: Monday 9/25/17

* Discussion/Meeting with Group Members

Start: Tuesday 9/26/17

End: Tuesday 9/26/17

* Individual Budget

Start: Wednesday 9/20/17

End: Tuesday 9/27/17  
Due: Monday 10/02/17

* Progress Report (Acquisition of Components)

Start: Tuesday 10/03/17

End: Thursday 10/12/17

Due: Monday 10/16/17

* Progress Report (Mechanical Assembly)

Start: Tuesday 10/17/17

End: Saturday 10/21/17

Due: Monday 10/23/17

* Individual PCB due

Start: Saturday 09/30/17

End: Saturday 09/30/17

Due: Monday 10/30/17

* Status Meeting

Start: Tuesday 10/31/17

End: Saturday 10/31/17

Due: N/A

* Group Placard

Start: Tuesday 10/24/17

End: Thursday 11/2/17

Due: Monday 11/6/17

* Individual Build Video

Start: Tuesday 10/3/17

End: Thursday 11/1/17   
 Due: Monday 11/13/17

* Individual Progress Report

Start: Thursday 11/9/17

End: Saturday 11/9/17   
 Due: Monday 11/13/17

* Hardware Demonstration

Start:

End:

Due:

# 7. Conclusion

This project has been created with the hope of reducing the drinking and driving rates and to lower the death rates caused due to this. This breathalyzer is intended to be a personal device with prospects to be introduced in bars or locales that serve alcoholic beverages. It is our sincere hope that this project will be used for the greater good. The final version of our project meets all of the specifications that were required to be completed.

# 8. Recommendations

The hardware that we’ve created can be improved upon in so many ways. Firstly beginning with the wiring; it can be done a little neater especially if the project was to be reproduced for consumer use. Currently we are using an adapter as our power source. That can be swapped with a portable charger. Another thing would be the cost of the sensors. The most expensive sensor in this project was the Polar Heart Rate Receiver and Transmitter. Most of the websites typically have these at the same prices but the other sensors can be bought in bulk for a much cheaper price. In the long run, it will cut your shipping cost. Once you’ve completed the project and it’s in working order, you can feel free to make any tweaks or improvements that you feel fit.

# 9. Bibliography

[1] Heartbeats in Your Project, Lickety-Split ♥. (n.d.). Retrieved February 02, 2018, from https://pulsesensor.com/

[2] #237238, M., #321089, M., #661774, M., O., #721982, M., #554862, M., . . . G. (n.d.). Pulse Sensor. Retrieved February 02, 2018, from https://www.sparkfun.com/products/11574

[3] #637052, M., & O. (n.d.). Alcohol Gas Sensor - MQ-3. Retrieved February 02, 2018, from https://www.sparkfun.com/products/8880

[4] Industries, A. (n.d.). Heart Rate Educational Starter Pack with Polar Wireless Sensors. Retrieved February 04, 2018, from https://www.adafruit.com/product/1077

[5] Institute of Electrical and Electronics Engineers. (2015, August 28). IEEE Xplore Digital Library [Online]. Available: https://ieeexplore.ieee.org/search/advsearch.jsp

[6] Wang, X., Jin, J., & Li, S. (2008, September 03). Measurement and analysis of heart signal based on the pressure sensor. Retrieved February 04, 2018, from http://ieeexplore.ieee.org/document/4618175/

[7] Malathi, M., Sujitha, R., & Revathy, M. R. (2018, February 01). Alcohol detection and seat belt control system using Arduino. Retrieved February 04, 2018, from http://ieeexplore.ieee.org/document/8275841/

[8] Kirtana, R. N., & Lokeswari, Y. V. (2017, June 08). An IoT based remote HRV monitoring system for hypertensive patients. Retrieved February 04, 2018, from http://ieeexplore.ieee.org/document/7944086/

# 10. Appendices