# **Task 2: Design of an app for the project**

**Technical Document**

**Outline**

* Discovery
* Story behind this
* Things used in this project.
* Affordable Digital Stethoscopes with AI
* Preparing
* Building the Stethoscope and recording sound
* Recording from Stethoscope
* Training TensorFlow Sound Classification AI
* Design
* Development
* QA
* Maintenance

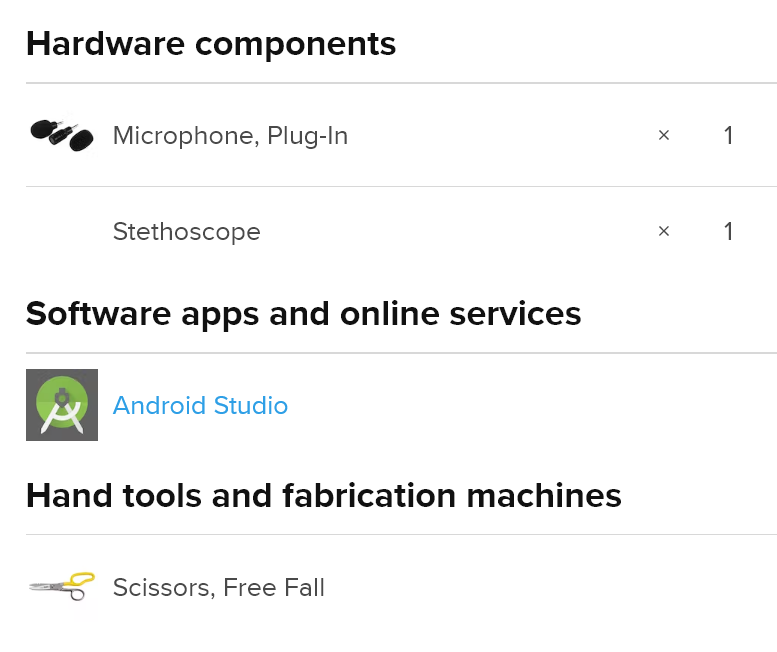
**Discovery**

Building a digital stethoscope under $1. Using AI for diagnosing respiratory symptoms, and empowering doctors via telemedicine for COVID19.

***Story behind this***

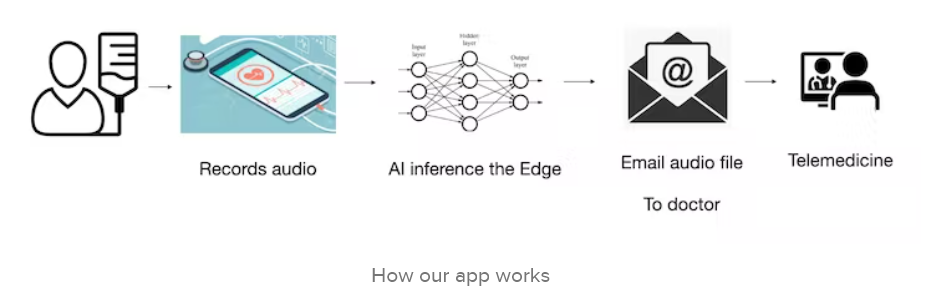
We all know COVID19 is extremely contagious by now, and shortage in healthcare provider is already happening in first world, this would only further escalate in 3rd world nations who do not have much resources. The healthcare industry continues to evolve and is adopting telemedicine to facilitate the accessibility of health care services. Every doctor has a stethoscope, but how many people own their own? And the digital stethoscopes on the market are generally not affordable at an individual level, especially in developing nations and struggling communities, they are a great tool for the doctor, but in telemedicine, we need the AI tool to be able to diagnose patients remotely that’s affordable at the same time. For this reason, we need digital stethoscope that’s under $1.00 accessible to everyone so that we can empower doctors to treat more people.

**Things used in this project**



**Affordable Digital Stethoscopes with AI**

In this project, we will also train an AI to use sound classification so that patients can do a self checkup anytime, and when the AI detects possible symptoms, the audio data is recorded for the patients to send it to a doctor for a further checkup.



**Preparing**

### **Step 1: Building the Stethoscope and recording sound**

To make this, we only need a normal Stethoscope and a tiny microphone, the Stethoscope can be purchased $5 retail, the Microphone for another $5 retail. Although this adds up to $10 retail, we do not need most of the parts. Also, if we were to order these whole sale, we can easily cut the entire bill of material down to about $1.



Now that we have the materials, we can cut the scope itself and attach directly to the small microphone. This would allow sound goes directly through microphone itself coming from the plate. Full building instruction can be seen below.

<https://www.youtube.com/watch?v=Ecgj3-3AWA8> [Making the Digital Stethoscope]

This step is enough for those simply want to make an affordable stethoscope and record your own heart rate or breath.

### **Step 2: Recording from Stethoscope**

Now that we have built a digital stethoscope from off the shelf material, we can use that to record our heart as well as our respiratory system. The recording can be done via our computer or our phone, making it much more accessible around the world

In Developing Nations, phones are much more common for recording, the 3.5mm is universal among all phones, for those who does not have it, a USB adapter can be very easily obtained.



When all said and done, you can see it like below.

<https://www.youtube.com/watch?v=fQJXtpvCzgs&feature=emb_imp_woyt> [Recording from Digital Stethoscope]

### **Step 3: Training TensorFlow Sound Classification AI**

Users now can record their own internal sound from Stethoscope, we will TensorFlow sound classification to determine whether it's COVID related such as Pneumonia, or other disease which you can wait for.

We can train this part on the cloud, for this article we can easily spin up a GPU server server using TensorFlow 2.2 on Google Cloud Platform.

This would have most of the prerequisite installed, and we can go straight into modification in the ubuntu system itself. We will be working with modified version of speech command example.

$ mkdir workspace  
$ cd workspace  
$ mkdir tmp  
$ cd tmp  
$ git clone https://github.com/tensorflow/tensorflow.git  
$ cd ~/workspace/tmp/tensorflow/tensorflow/examples/speech\_commands

To check TensorFlow is properly installed

import tensorflow as tf  
print(tf.version.VERSION)

As for data, we can retrieve data from different sources. For this project we will be able to classify 5 different types of sound. To make the project simple, we will focus on Trachea, except for heart rate, which we will be using stethoscope on the heart.

* Healthy
* Chronic Obstructive Pulmonary Disease
* Pneumonia (often associated symtom with COVID19)
* Upper Respiratory Tract Infection

We have to change [train.py](http://train.py/) to following format

We can load the files into ****training\_data**** folder and, change the training file to following

$ python3 train.py

After a few hours we should have our models ready to use

We can launch tensor-board to take a look at the data, currently you'd have to downgrade to TensorFlow 2.1.0 in order to load this, as time of this writing the TensorFlow 2.2.0 is having some problems launching tensor-board, I've already filed a report.

Additionally, we can use tensor-board.

tensorboard --logdir retrain\_logs/

After training now we can freeze the model via

python3 freeze.py \  
--start\_checkpoint=training/conv.ckpt-18000 \  
--output\_file=model/froze\_graph.pb

**B) Design**

Now it’s time to give your App its own unique look and feel. Making concept of App by Sketch, Wireframe and Prototype. Clickable model of the app that looks real app.

1. **Development**

Now the design is ready, but it’s still a lot of work to turn a model into a fully-functional product. With limited team and time, I would like to develop this app in Open Source Cross Platform Technology (Flutter) to target all the users either they use Android, IOS, Web, or Desktop. What is more economic then using single code base to target multi-platforms. It saves development and maintenance cost and time.

The development process can be divided into two parts.

* **The front-end**. This is the face of the program with which users interact. The task of a front-end developer is to guarantee a flawless user-friendly experience using Flutter, Android Studio.
* **The back-end.** This is the functional part, the server-side of the application. Using TensorFlow, Firebase real-time DB. We will use Audio classification Tflite package (https://pub.dev/packages/tflite\_audio/versions/0.2.1) for flutter (iOS & Android). Can support Google Teachable Machine models. Tutorial can be teach here (https://carolinamalbuquerque.medium.com/audio-recognition-using-tensorflow-lite-in-flutter-application-8a4ad39964ae).

In this stage, we will have prepare first version or MVP (minimum valuable product).

1. **Quality Assurance**

In this stage we will test the app, fix bugs if any, do unit testing to avoid negative code.review.

1. **Release to the store**

Finally, you can publish the application to the store. Google play store or App store.

1. **Maintenance**

In this stage, we keep polish our app by getting user’s reviews and suggestions or adding more advance features.