The Development and Deployment of Mobile Apps and Server Platform for Real-World Screening of Pulmonary and Cardiovascular Disease in Low-Resource Areas

by

Vedaant Kukadia

B.S. Electrical Engineering and Computer Science, Massachusetts Institute of Technology (2020)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2021

Author . . . .
Department of Electrical Engineering and Computer Science
August 6, 2021

Certified by . . . .
Dr. Richard Fletcher
Research Scientist
Head, Mobile Technology Lab
Thesis Supervisor

Accepted by . . . .
Katrina LaCurts
Chair, Master of Engineering Thesis Committee

The Development and Deployment of Mobile Apps and Server Platform for Real-World Screening of Pulmonary and Cardiovascular Disease in Low-Resource Areas

by

Vedaant Kukadia

Submitted to the Department of Electrical Engineering and Computer Science
on August 6, 2021, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

In order to address limited access to health care in low-resource parts of the world, our group at MIT, over the past 7 years, has developed a variety of health screening tools that rely on a smartphone with access to a remote server. This mobile health platform, known as "PyMed," consists of a Django server, Postgres data base, and a variety of Bayesian network machine learning and data processing algorithms implemented in Python. While several different server platforms have been demonstrated by our group over the past few years, a great deal of additional development was required to deploy these technologies in a real-world scenario. In addition to the mobile application software and machine learning algorithms, actual deployment of these technologies required the development of transaction sequences and work flows that enable a health worker, or doctor, in the field to collect data from a patient, process the result in real-time on the server, and then receive a complete and usable result on the mobile phone. In this thesis, I discuss the detailed workflow and underlying technology required to perform diagnostic health measurements in a real-world setting. I present the various software modules and server API work that needed to be developed. In addition, I describe how the health results were designed and ultimately presented in a simple and usable format that both the patient and health worker could use and understand. In this work, I describe two disease categories: cardiovascular disease and pulmonary disease. In total, our group has developed two separate servers, 11 mobile apps, and multiple algorithms for signal processing and diagnostic prediction for these two disease categories. The work in this thesis was completed in preparation for several field studies in Bangladesh: (1) a study with coronavirus patients in the NIDCH Hospital in Dhaka, Bangladesh; and (2) an efficacy study with private community health workers in two low-resource areas of Chittagong District and Jamalpur, Bangladesh.

Thesis Supervisor: Dr. Richard Fletcher

Title: Research Scientist
Head, Mobile Technology Lab

Acknowledgments

I would first like to thank my advisor, Dr. Richard Fletcher, for his continued support and guidance. Dr. Fletcher's work to revolutionize and make a real impact in global health has inspired and motivated me in many ways, not only with regards to my work. I am thankful that I was introduced to the field, and hope the work presented in this thesis will have a real impact to help those in low-resource settings.

I would also like to thank the entire group of the MIT Mobile Technology Lab, for their assistance and guidance in the project, with a special thanks to Saadiyah Husnoo for offering her continued support and guidance throughout.

Finally, I would like to thank my parents, Dr. Ashok Kukadia and Dr. Baishali Paul Kukadia. They have always put the interest of their children first, and I would not have made it to this point without their encouragement, support, and love. Lastly, I would like to thank my two sisters, Saaanchi Kukadia and Khushi Kukadia for always supporting me and encouraging me to put my best foot forward in everything I do.

Contents

Chapter 1

Motivation: Global Health Needs and Mobile Platforms

1.1 Introduction

While healthcare overall has been advancing at unprecedented levels, patient care in low-resource settings remains a global challenge. This is due for a variety of reasons, including a lack of existing and available medical infrastructure, lack of medical tools and hospitals, as well as a very low number of trained healthcare professionals. The difficulty in mitigating these limitations lay in the fact that there simply aren't enough tools out there to support healthcare in low resource settings. While groups are working on bridging the healthcare gap between the developed and under developed communities, the tools offered are usually over simplified, and only work well under the most ideal conditions, which are often unattainable.

In order to improve the health outcomes in low-resource settings, it is crucial to increase the number of healthcare professionals. It is also crucial to maintain a standard of care that ensures equal access to all. While this is a long term goal, the aims of our group are to empower clinicians and health care workers with easily accessible and cost effective tools to screen for and identify those at highest risk for chronic diseases and to help ensure that those individuals seek out a treatment plan.

1.2 Health Camps

Health camps can be an effective and convenient short-term solution for healthcare treatment and diagnosis in developing countries. These camps are usually initiated by nongovernmental organizations and political organizations [1]. Health camps, however, tend to be disease specific and very dependent on need. Organizations sponsoring these camps will unlikely target areas without substantial need, making these camps unreliable on a large scale. Setting up multiple health camps treating different diseases can help alleviate some of these issues, but poses new risks as now primary care physicians, or general practitioners, are needed in order to guide patients to the correct path of care.

1.3 Community Health Workers

To help facilitate the need for more healthcare workers, many developing countries have integrated community healthcare workers into their healthcare system. While community healthcare workers lack formal medical training, they have expertise in fundamental care that can be critical for saving lives. These workers act as a means of connecting the healthcare system of developing nations to rural populations, where they provide first aid, maintain medical records, and mobilize the community when needed [2].

In India, there are currently around one million Accredited Social Health Activists (ASHA), who have been trained through the ASHA program, with the aim of increasing community engagement with the health system and supporting access to public health services [3]. The recent increase in the availability of, and access to, smartphones globally has created a new means for delivering cost-effective mobile diagnostic tools to low-income regions. Recently, the Indian government has began to distribute smartphones to ASHA workers, allowing these individuals to use technology for advances in treatment.

1.4 The Need for Mobile Screening Tools

As much of our everyday lives transition online, many of the services offered have also moved virtual. For example, many doctors are opting for telehealth appointments over in-office appointments. While this is a small step in integrating healthcare with technology, the potential that technology has to transform the way we prevent, treat, and diagnose different diseases is just at the beginning. This is especially true for low-resource settings where access to healthcare is severely limited.

There have been many studies that focus on low-cost, effective models for screening and diagnosing of patients for a single disease. Dozens of mobile phone platforms exist for use in public health, including Dimagi, D-Tree, and MedicMobile. The development and deployment of these mobile applications is forming a larger field known as mHealth. While development in mHealth is increasing, the use of AI/-Machine Learning is essentially very limited. Additionally, efforts have mostly been focused on deploying ML algorithms in developed countries, with little work being done to promote it's use in low-resource areas. [4] This is in part due to a lack of easily accessible centralized and labeled data, but more importantly to the absence of health workers to collect such data, and the absence of cost-effective tools to measure and screen for diseases. Given recent developments in mobile computational power, and an increase in connectivity to remote servers enabling mobile devices to perform even more computationally intensive tasks, we explore a relatively new pathway for machine learning based diagnoses.

1.5 Challenges with Cloud-Based Solutions

Although the potential for cloud based solutions remains very promising, in a practical setting it is difficult to deploy a system due to the fragmented nature of the technologies and healthcare data needed. For example, a reliable integrated client-server platform requires a secure server architecture, mobile applications, and machine learning. Thus, development of such a system requires a multi-disciplinary team. As a result, many of the current solutions are developed by large companies, as they maintain the infrastructure to support and iterate on these systems. Additionally, they also have access to consumer data through their smartphones, smart watches, and other products offered.

Other commercial companies entering the healthcare space include FitBit and Whoop, however, the services offered are very limited in the sense that they provide primitive health statistics, which don't have many applications in terms of clinical care. These devices are also costly, limiting their use in low-resource areas.

1.6 Current Global Health Needs

In order to support the global need for an effective mobile health system, there are a few key components that need to be addressed. Mobile applications must be able to collect patient data, and connect to remote servers in order to run machine learning algorithms. To support machine learning development, researchers need access to patient data and clinical labels provided by healthcare professionals. Additionally, this data must both be accurate, and correct. To allow for practical use, clinicians must be able to store their patients information in a secure and private manner. For personal use, diagnostics and results must be conveyed to the user in a way that is easy to understand and visual. Lastly, to support third-party integration the system must be robust in order to comply with local government rules and regulations. These functions are further outlined below:

1. **Data Collection**

   The system must be able to support data collection both in a field test and in a clinical setting. Mobile devices must be able to collect and store data until there is service, and then upload that data to a remote server. Additionally, because the system is handling sensitive and private information, the server architecture must be secure and reliable.

2. **Labelled Data**

In order for machine learning algorithms to be trained and used effectively, the system must have in place the ability to label data. In other words, clinicians need to be able to access various measurement data and supply clinical labels at both the measurement level, as well as diagnosis level. Because a diagnosis can be based off of many measurements, it is important to label the measurements accordingly if it contributes to a diagnosis.

3. **Remote Server Integration**

   The system must be able to access machine learning models and other algorithms in real time. Once a connection is established, data is sent from the mobile device to the remote servers, and the client waits for a response displaying either an analysis or a diagnostic response. This remote server integration must be reliable, fast, and secure so that no patient information is leaked.

4. **Clinician Support**

   As one of the use cases for the system is for clinician use, clinicians must be able to store patient information and data. This is similar to an electronic medical record system, or EMR.

5. **Understandable and Visual Results**

   Another use case is for community health workers, or even patients themselves. Because they may not have former medical training, it is necessary to relay results and information in an easy to use and understandable way. Additionally, the process of recording and saving measurements must be intuitive, diagnostic results self-explanatory and clear, and the ability to save and share results simple.

6. **Third-Party Integration**

   For commercial use, there are a few considerations that must be met. First, in many cases we would like to extend our ML algorithms to other research partners without revealing the implementation. Additionally, we may want to share our EMR and data collection functionality without revealing any of the algorithms powering our mHealth system. Lastly, it is important to be able to configure the system to work with multiple different servers so as to follow and adhere to local government regulation. For example, in some developing countries patient data is restricted from leaving the country. To work around this, our system must be customizable to work with local servers.

7. **Offline Support**

   Many of the algorithms used in an mHealth system require access to remote servers, however, in low-resource areas it may be the case where no internet connection is available. In this situation, the system must be able to operate offline, and for and analysis that require remote server access, those operations are stored on the mobile device until a connection is established. This ensures continuous and uninterrupted use in the event of connection loss.

Chapter 2

Existing Solutions

2.1 Introduction

The healthcare ecosystem is rapidly progressing towards a more digital form, and we are seeing more and more mobile cloud based platforms enter the scene. Most of these platforms, however, are not designed to screen and diagnose for different diseases, and do not employ any machine learning algorithms. Instead, they tend to focus on one aspect of healthcare, for example tracking motion, analyzing behavioral trends, measuring sleep patterns, etc. but fail to treat health holistically.

2.2 Existing Cloud Based Platforms

A recent study performed by Wang *et al.* searched through PubMed for mHealth-related studies on diabetes and obesity treatment and management published since 2000, and found that mHealth solutions tended to focus on three things: mobile phone text messaging, wearable or portable monitoring devices, and smartphone apps. [5] The main use case for these mHealth services is for personal use, or for people who want a way to track their motion and trends in order to intervene and change the course of their future. They are not specifically designed to diagnose and screen for diabetes, and don't contain the infrastructure to maintain patient data on a larger scale.

Another platform, AlemHealth, offers various mHealth solutions to those in low and middle income countries. One of the services offered is a telecardiology screening program that provides battery-powered 12-lead ECG machines, at a fraction of the cost of leading alternatives.[6] Readings, however, are sent to cardiologists who then interpret and read the ECGs in order to provide a diagnosis and treatment plan. This poses a new challenge: the testing site needs individuals who can properly operate these machines, and then readings must be sent to a professional to get a diagnosis.

Other platforms, such as OpenMRS, which is an open-source system, focus on facilitating global telehealth by allowing users to build their own customized solutions. [7] Another service, D-Tree, empowers health workers with digital tools to better collect information in the form of a questionnaire. This way, the service can provide treatment plans and early screening to those at risk. [8]

2.3 The Need for Improved Cloud Based Platforms

While all of the above platforms have drastically improved health outcomes for those in low-resource areas, they have certain limitations. Most platforms are not holistic, meaning they aim to assist in one specific aspect, but cannot be used standalone to diagnose and screen for diseases. For example, they may need a health care worker or clinician to review responses collected by these applications before making a diagnosis. Thus, the data collection step and decision making step are done at different times. In a setting where healthcare is very limited, it may be hard to identify and track down patients at later times to inform them of their diagnosis. There is a large opportunity to leverage the power of machine learning algorithms to diagnose and screen patients in real-time.

To support the AI algorithms and machine learning models in a clinical setting, better server architectures are needed than what currently exists so that the potential of these algorithms can be fully utilized. For example, many of the platforms discussed above are used privately. While they collect valuable data, and in some cases even have actual health care professionals provide clinical labels, these data points are not being used to train machine learning models. Collecting various measurement types, and feeding them into a machine learning model has the potential to drastically change healthcare at a global scale.

Chapter 3

Prior Work in Our Group

3.1 Introduction to PyMed

Over the past few years our group at MIT, the Mobile Technology Lab, led by Dr. Fletcher, has been working on filling in the gap between existing health cloud platforms and current global needs. In 2019, graduate student John Mofor developed the initial version of this architecture, coined the PyMed server. The principal behind this client server architecture was to allow for the integration of AI algorithms paired with a cloud based EMR system to be used in a real world setting.

The initial version of the PyMed included an electronic medical record system that kept track of patient data, and was paired with a set of mobile apps used to record measurements. This architecture allowed for centralized medical data collection and real-time analysis, and has been deployed in several settings. Clinicians and health workers can use the mobile apps to collect data with or without internet connection, and when connection is available, the data is uploaded to the cloud and an analysis can be performed in real time.

To better understand the system and how it operates, we provide a brief description of some relevant concepts.

3.2 Users and Usergroups

There are three different roles that exist in the PyMed server, each with their own set of permissions: admins, clinicians, and patients. Admins are the "super" users and can manage the permissions of all other users in the system. Clinicians manage data, collect measurements for patients, and can provide labels for both measurements and for patients. They maintain study groups, and can add patients to these. Lastly, patients are the subjects in a study. They are the individuals who we aim to screen and diagnose, and their data is usually managed and collected by clinicians.

In this thesis, we discuss an anonymous mode setting in our Android applications that will allow patients to collect their own data and run our measurement and diagnostic algorithms on that data without needing to register with our EMR system. This will be further explained in Chapter 7.

3.3 Measurements

Measurements include the specific type of data we would like to collect on a patient. For example, a measurement may be a cardio questionnaire, a lung sound recording, or pulse waveform recording. These measurements are recorded using our measurement applications. In the PyMed system, each measurement corresponds to exactly one entry in the `DiagnosticMeasurement` table. Additionally, each measurement is also associated with some metadata, and this is stored in a separate table known as `DiagnosticMeasurementMetadata`. The metadata of a measurement includes useful information such as:

1. Duration: the length of time taken to perform the measurement

2. GPS Location: The coordinates at which the measurement was taken

3. Patient ID: The ID of the patient the measurement corresponds to

4. Clinician ID: The ID of the clinician who recorded the measurement

5. Timestamp: The time at which the measurement was taken

3.4 Screening vs Diagnosis

It is important to make the distinction between screening and diagnosis. Screening is typically done on patients without symptoms, and is used to identify patients that seem healthy, but may be at risk of having a disease. These tests are usually routine. Diagnostic tests, on the other hand, are used to determine what disease a patient may have after showing symptoms. Additionally, diagnostic tests may be performed after a screening identifies some abnormality.

3.5 Real-time Diagnostic Algorithms

Some measurements in our system involve processing before they can be understood and utilized by our diagnostic algorithms. For example, measurements recorded by the PPG Camera app and PPG Berry app used in the cardio projects, consist of a pulse waveform. This waveform must be sent to our server for further analysis before we can extract information such as the patients heart rate variability, heart rate, pulse wave analysis score, etc.

To support the ability to run real-time algorithms, we have two main APIs: the `run_analysis` API and the `process_measurement` API. The `process_measurement` API is intended to run algorithms that correspond to measurement data. That includes the above described PPG Berry and PPG Camera measurements. Not all measurements require further processing, however. For example, questionnaire data can be used in our diagnostic algorithms as is. The `run_analysis` API is intended to be used to run our diagnostic algorithms. On the cardio server and pulmonary server, the diagnostic algorithm is a Bayesian network that takes as input the results recorded from our measurement applications, and outputs a diagnostic report to the user. This report can be downloaded, shared, and used as a means to seek further medical treatment if an abnormality is identified.

3.6 Mobile Applications

One of the most important elements for user-engagement with our services is the ability to interact with our server and algorithms through an intuitive user interface. Recently, our lab has begun to develop mobile applications to allow for this. Through our applications, clinicians can monitor data, view patient records, download and upload data from and to the server, as well as run our machine learning algorithms. Because the handling of data is crucial to the usability of our applications, special attention was paid to ensuring the apps were as intuitive as possible, and the database schema was easily understandable and maintainable.

3.6.1 Measurement Applications

Measurement applications are used to record measurements and can either be accessed on their own, or through the container application, which is described in the next section. For the cardio project, we currently have three measurement applications: PPG Camera, PPG NAJA, and Cardio Questionnaire. In Chapter 7, we discuss the addition of a fourth measurement application to to the cardio screening apps: PPG Berry. A view of the measurement apps from the container app is shown in 3-2.

3.6.2 Container Applications

Each project has a container application. Container application group the measurement applications needed for a particular disease screening and launches only the measurement applications you need. It is to be used by clinicians or health care workers who are collecting data and using our algorithms. It is essentially an interface for our EMR system. Clinicians can maintain patient records, add diagnoses, provide clinical labels, view past history, as well as interact with our measurement apps, as shown in figure 3-1. There are two modes: clinical mode and anonymous mode. Clinical mode is used by clinicians who require patient information to be saved and uploaded to our servers. Anonymous mode is intended for cases where patient information must be kept private, and the use case is instead for screening patients

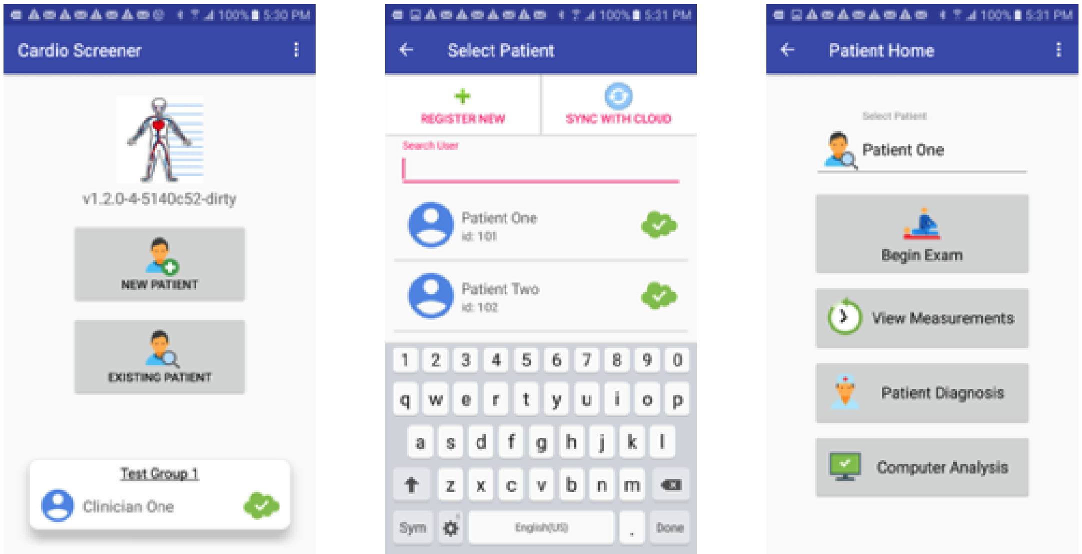


Figure 3-1: Screenshots taken from the Cardio Screener mobile application. Existing and new patient registration (left), list of current patients in EMR system (middle), and patient homepage (right)

for disease as opposed to maintaining patient data and information. Measurement apps can either be launched from the container app, or they can be used on their own.

3.7 Work Flow Example

Here, we describe the basic workflow for taking and analyzing a generic measurement using our mobile applications.

1. **Log in** If you are a clinician operating in clinical mode, here you are tasked with logging in to the PyMed EMR system. This is implemented in the Android application, together with the remote server database.

2. **Register The Patient (optional)** If operating in clinical mode, a patient is registered to a study. This is implemented in the Android application, and the information is then sent to the remote server database.

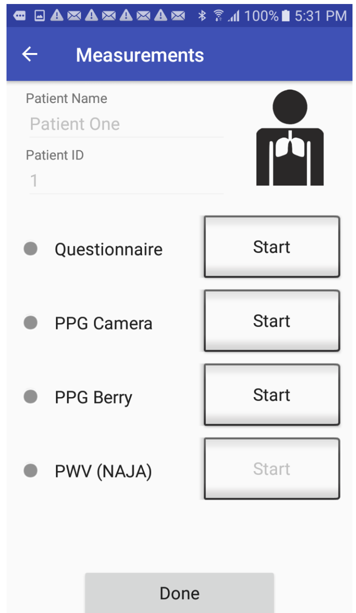


Figure 3-2: Available measurement applications accessible from the Cardio Screener mobile application

3. **Administer tests and collect data from patient** Implemented using one or more of the measurement mobile applications, such as the questionnaire app, recording a sound from a stethoscope, or recording an image.

4. **Process the data that was collected to check for quality** Data processing is handled on the remote server. If data is not of quality, it may be recollected.

5. **Request a diagnostic algorithm to be run on the phone or on the server, using one or more measurements** For local analysis, this is done on the Android device using a logistic regression model. For a cloud analysis, this is done on the remote server using a Bayesian model.

6. **Run the diagnostic algorithms on the server, and include error codes if needed** The diagnostic algorithm is run on the server, and returns either a diagnostic ID in the case of a success, or an error code.

7. **Receive and Display the results of the diagnostic algorithms** If successful, the diagnostic algorithm returns a diagnostic report in the form of an HTML

page. This is generated on the server, and displayed on the mobile application. For use by both patients and health care workers, the output has been designed to be easily understood, even by individuals with no medical training. As such, values are color coded to indicate health.

8. **Patient or health worker can package the Diagnostic report and forward it to another doctor or family member.** This is done on the Android device. The report can be saved as a PNG file, and shared with others.

9. **The results can be logged on the server along with a map visualization for use by NGO's and government organizations.** Diagnostic results generated are saved on the server, and can be visualized using our web application.

3.8 Shortcomings of Current PyMed Platform

In global health, the ability to screen patients for disease while also following local government regulation is of critical importance. Because many governing bodies require patient data to be confined to within the country, as well as patient privacy and health to remain within local regulation as opposed to third-party services, our current EMR system is unusable. For our group, this meant maintaining an anonymous version of our screening tools, where patient data is never saved. Rather, health workers can collect data using our mobile applications, and then simply run our diagnostic and measurement applications without having to register patients in the PyMed EMR system.

As mentioned previously, the current version of the PyMed server lacked an anonymous version. While recent graduate, Saadiyah Husnoo, built out the server architecture and APIs to support anonymous mode, this was not integrated into our mobile applications.

Chapter 4

Overview and Thesis Scope

4.1 Project Overview

In this project, we aim to build on the efforts and work of previous researchers in the Mobile Technology Lab at MIT to deploy our tools in a field test environment. We develop a system that supports multi-disease screening used by health workers in rural clinics and health camps, while addressing practical challenges related to patient privacy, foreign government regulations, and intellectual property concerns.

The early chapters focused on setting the scope of mobile health in a global setting, with a focus on low-resource areas. We also discuss existing mobile health solutions and current needs of low-resource area, as well as why these solutions are deficient, and what work needs to be done to bridge the gap between what is needed and what exists. Lastly, we outline previous work done in the Mobile Technology Lab at MIT, and the foundations upon which this work was built. This includes the development of an electronic medical record, or EMR, system, the necessary server and architecture work to support such a system, and the disease-specific algorithms that power our tools.

Chapter 5 explains cardiovascular diseases and the effect it has on lives globally. We discuss traditional screening methods, and then describe our general approach to CVD screening and diagnosis. We touch upon three main measurement types used for screening: framingham risk score, PWA, and PWV.

Chapter 6 discusses the algorithms and machine learning models used to diagnose and screen for CVD. We introduce a Bayesian model, based off of a similar architecture to the model built by a previous student in our lab, Aneesh Anand, to diagnose pulmonary disease.

In chapter 7, we describe the mobile app development work done on the cardiovascular projects, namely, connecting the various algorithms built by previous students into a single mobile solution, as well as building out functionality for an anonymous mode. We also introduce a new measurement type to our screening tool, the PPG Berry app.

In chapter 8 we discuss the necessary server development work needed to support our cardiovascular screening apps. We discuss the work done by previous graduate student, Saadiyah Husnoo, as well as further contributions made to allow for the mobile apps to be used in a field test.

Chapter 9 moves on the the work done for pulmonary diseases. We discuss the burden of pulmonary diseases globally, traditional screening methods, and then describe in detail our general approach to pulmonary screening and diagnosis.

In chapter 10 we briefly discuss the algorithms and machine learning models powering our diagnostic result, and in chapter 11 move on to describe mobile app development. This includes the introduction of a blood test questionnaire, as well as modifications to our previous questionnaire to address needs imposed by the COVID-19 pandemic. In chapter 12, we discuss server development done for pulmonary disease screening, and what future work is necessary to complete the anonymous mode.

Lastly, in chapter 13 we discuss the challenges of deploying such a system in the real world. This includes setting up various servers to comply with foreign government regulations, enabling anonymous mode of our apps to ensure patient privacy and safety, as well as multi-disease support and integration. Due to the current circumstances, this section also describes the added need for COVID-19 support.

Chapter 5

Cardiovascular Disease Screening

5.1 Cardiovascular Disease

Nearly 18 million people die each year from cardiovascular diseases, according to the World Health Organization, and this accounts for roughly 32% of all deaths worldwide. Cardiovascular diseases, or CVDs, encompass a group of disorders of the heart and blood vessels. Common classifications of CVDs include hyptertension, atherosclerosis, cardiomyapathies, and arrhythmias.

While CVDs can affect people from all different backgrounds, their presence in low-resource areas is especially high. Nearly 80% of CVD deaths occur in low and middle income countries, with hypertension as the most notable risk factor. [9] Other notable risk factors that contribute to heart disease are an unhealthy diet, lack of physical activity, use of tobacco, and use of alcohol.

5.2 Diagnosing and Screening Cardiovascular Disease

Traditional methods for diagnosing CVD include various tests including blood tests, a chest X-ray and a physical exam, an assessment of personal history as well as an assessment of family history. Other tests used to diagnose heart diseases are shown in

Test	Description
Electrocardiogram (ECG or EKG)	An ECG is a test that records the electrical signals in the heart, and can be used to detect abnormal heart rhythms. It is quick and painless.
Echocardiogram	A noninvasive exam that uses sounds waves to produce detailed images of the heart's structure. It shows how the heart beats and pumps blood.
Stress Test	This test involves raising the patient's heart rate with exercise or medicine while performing heart tests and imaging to measure how the heart responds.
CT Scan	In a cardiac CT scan, the patient lays on a table inside a circular shaped machine. An X-ray tube inside the machine rotates around the patients body and collects images of the heart and chest.
MRI	A cardiac MRI uses a magnetic field and computer-generated radio waves to create detailed images of the heart.

Table 5.1: Tests to diagnose cardiovascular diseases [10]

table 5.1.[10] These diagnostic tests are not available to physicians in many parts of the developing world, due to high costs for the equipment as well as need for specialized technicians to operate and maintain these machines. Furthermore, in areas with a limited access to healthcare professionals, even if the equipment is present there may not be specialists trained to diagnose for specific cardiovascular diseases.

Tools for diagnosing cardiovascular disease are generally very limited in low-resource settings. Sometimes, these tools can simply be a clinical history questionnaire that contains questions the patient doesn't know the answer to, proving them even less effective as diagnostic tools.

5.3 Challenges of Detecting Cardiovascular Disease

In the developed world, great strides have been made to reduce CVD risks. These changes include life-style interventions, and medications including antihypertensives, however, access to these medications in low-resource settings, despite being low-cost, is difficult and rare.[11] While many of the risk factors that contribute to CVD can be mitigated by behavioral, cultural and societal changes, it is difficult to implement these changes without proper medical guidance and instruction to those in the area.

Unlike many diseases, CVD diseases can often be asymptomatic. To screen and diagnose for them, it becomes necessary to have the proper tools to measure blood pressure, cholesterol levels, and other biochemical parameters. In low-resource settings, there are two main limitations to predicting CVD risk: the first being a lack of the proper tools to obtain these measurements, and the second being the fact that many of the risk assessment scores were developed for populations of European descent, and individuals in high-income areas. Translating these risk metric scores over to low-resource settings is often incomparable. Therefore, it is important to develop low-cost, non-invasive, efficient tools to screen and diagnose individuals at risk for CVD.

5.4 Our Approach to CVD Screening

In order to screen for cardiovascular diseases, our group has developed a holistic mobile platform with three main measurement types: framingham risk score, PWA Analysis, and PWV Analysis. In this paper, I introduce a fourth measurement, PPG Berry, which is further explaiend in chapter 7. These measurements are collected using our mobile apps, and the resulting data is stored in our EMR system. Measurements are then sent to a remote server for analysis, and a Bayesian machine learning model outputs a diagnosis back to the client side.

5.4.1 Framingham Risk Score

The first measurement type used in our screening platform is the Framingham Risk Score. The Framingham Risk score was developed as a scoring method for assessing the risk of CVD by Wilson *et al.* and remains one of the most popular and widely used questionnaires. [12] We developed an Android mobile application based off of the Framingham model which asks patients to fill information including age, gender,

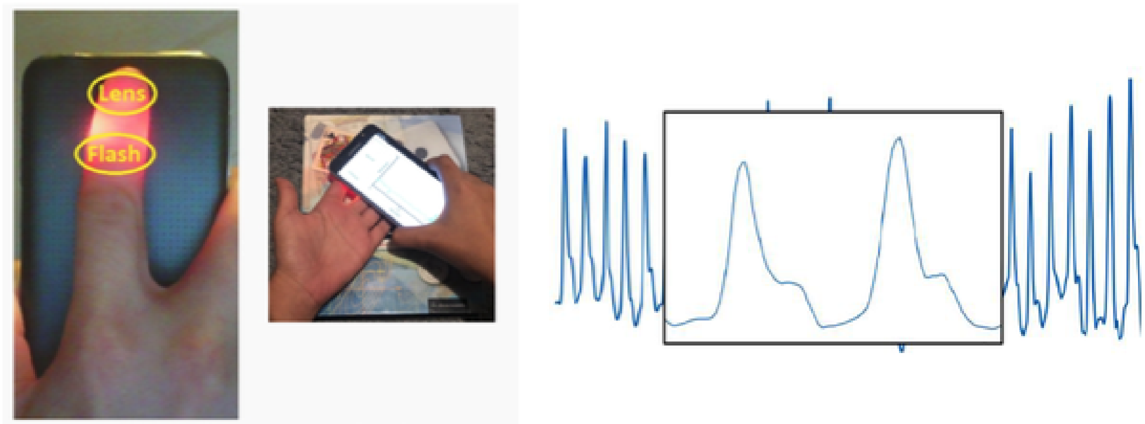


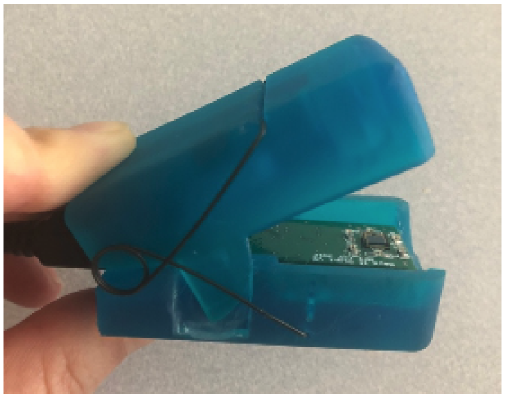
Figure 5-1: PPG Camera mobile application (left, center) and a sample PPG waveform (right)

blood pressure, blood cholesterol, diabetes diagnosis, and smoking frequency. This information can be collected by either a patient or a clinician, and the output of the model is a prediction of the patient's ten year risk of developing CVD. While this questionnaire is very useful, it is limited in the sense that it is difficult to obtain the answers to many of the questions in a low-resource setting. For example, in order to fill in blood cholesterol level, the patient must receive a blood test. For the purpose of our CVD screener application, we couple the Framingham Risk Score with other measurements that are more easily obtainable.

5.4.2 PPG Pulse Wave Analysis (PWA)

The second measurment type is a PWA analysis performed using a PPG waveform. Previous students in our lab, such as Josef Biberstein, have developed a smartphone-based mobile application that uses the phone's flashlight and camera to extract a PPG waveform from the finger. An illustration of this process is shown in 5-1 along with a sample collected PPG waveform.

Once the PPG waveform is collected using the PPG camera application, the raw data is sent to our servers. Our server then retrieves the raw data and runs the PWA algorithm. The results of the analysis are then fed into the bayesian algorithm which outputs a diagnostic result.



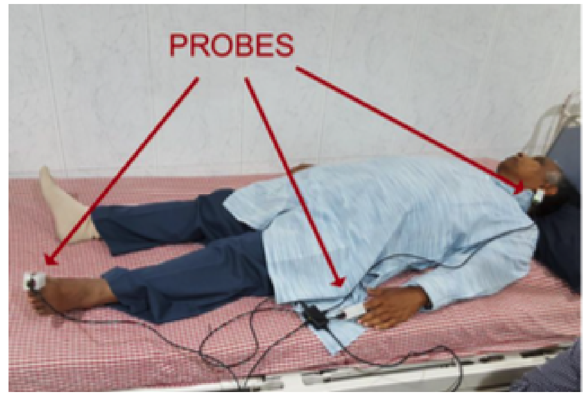


Figure 5-2: NAJA probe (left) and illustration of PWV measurement being taken (right)

5.4.3 PPG Pule Wave Velocity (PWV)

In addition to developing low-cost methods for extracting the PPG waveform from a patient, our group has also developed low-cost probes used for measuring PWV. The device has been named NAJA, and is comprised of three probes and a central interface. To conduct a measurement, a probe is clipped to the patients ear, finger, and toe. The central interface is then connected via a wire into a smartphone. We can then calculate the patient's PWV. An illustration of a probe and this process is shown in 5-2.

Due to the time frame of this thesis, the PWV algorithm was not used in the final implementation for CVD screening.

5.4.4 Screener Application

Lastly, our group has developed a container mobile application to encompass each of the three measurement apps. This app is used to store patient information and measurements, and is essentially used as an EMR. From here, a clinician can label measurements, conduct new measurements, run our Bayesian machine learning model to generate a diagnostic result, or input their own clinician diagnostic result. Any labels provided by the clinician, either for a specific measurement or a diagnosis for the patient, is then used as a training sample on the algorithm to further improve the accuracy of future results.

Chapter 6

Cardiovascular Disease Algorithms and Machine Learning

6.1 Prior Work

In prior iterations of our cardiovascular screening tool, we used logistic regression as our modeling approach. Logistic regression was chosen for it's simplicity, and low-computational requirements, meaning it could be run directly on a smartphone. This is very useful in low-resource settings, as the model can output a diagnostic prediction with or without connection to our remote servers.

Logistic regression involves learning weights for feature in inputted into the model. Those weights then determine how positively or negatively a specific feature is in determining the outcome the outcome, or diagnostic result. Logistic regression models are often used to predict the likelihood of a disease, however, these models have limitations. For example, they are mainly influenced by sample size, and they do not handle missing data well. [13] In a clinical setting, missing data is very common, and so a model that requires all data present will not work in all cases. Additionally, the parameters and weights obtained from a regression model are not well adapted to patients coming from different settings and populations. To address these limitations, a previous graduate student, Aneesh Anand, developed a Bayesian model to predict pulmonary diseases. Here, that model was extended and adapted to predict cardiovascular diseases.

6.2 An Overview of Bayesian Networks

Bayesian networks are acyclic directed graphs that use Bayesian inference for probability computations. In this network, nodes represent random variables, and arcs between nodes represent probabilistic dependencies between those variables. For example, if we have a directed edge from node A to node B, then A is considered to be a parent of B, and likewise, B is a child of A. The probability of a variable is then determined by conditional probability tables, given the parents of that variable. For variables with no parents, a prior probability distribution is used.

In our case, Bayesian models can overcome many of the limitations of a regression model. For example, Bayesian networks can process both quantitative data (heart rate, PWA Score) as well as qualitative data (gender, is the patient a smoker). Another advantage of a Bayesian model is that there is no *a priori* hypothesis about the nature of the modeled relationships.[13] Because a Bayesian model can also handle missing data, whether that be during training or when run to generate a diagnostic result, its use cases in a low-resource setting are very promising. Lastly, the model's performance can be enhanced with each new case input into the model. For example, if a clinician takes measurements and then provides a label for those measurements or a diagnosis for a patient, this can be used to improve the predictive quality of the model in order to fit that direct setting. This is beneficial when running the algorithm between clinical centers that may have different populations, but also within a clinical center if the characteristics of a population change over time.

6.3 Cardiovascular Bayesian Network

The final step in our cardiovascular disease screening tool is to output a diagnostic result to the client. Previously, the logistic regression model used a binary classifier to determine whether a patient was "healthy," or "at-risk/unhealthy." The developed

Bayesian model broke up the "at-risk/unhealthy" category into three new categories, "pre-CAD", "CAD", and "Hyptertensive." Using the results from the three measurement types: the framingham risk score, PWA score, and PWV results, a diagnostic result is output. The model has been trained on data from 68 patients. The breakdown of patients is as follows: 21 CAD, 11 preCAD, 5 hyptertensive, and 31 healthy patients. As of this writing, we do not have training data for the PWA score, and PWV is not used in the model due to the time frame of this thesis.

6.4 Heart Rate Calculations

A previous graduate student in our lab, Victoria Ouyang, developed much of the PWA algorithm currently deployed on our serer. To further extend her work, we added heart rate calculations. To calculate the heart rate from a pulse waveform, we use the following algorithm. In each 30 second recording, we identify the time at which a peak was recorded as T_n, where T is the time and n is the number of the peak. We then calculate the time delay between successive beats, or IBI for inter-beat interval. In other words, $IBI = T_n - T_{n-1}$. To convert each value, T_n, into from milliseconds to seconds we divide by 1000.

Because it is likely that we have multiple pulse peaks, we want to use the median heart rate. To calculate this, we take the median IBI value and divide it from 60. Note, that if any IBI value is greater than 1.5 seconds or less than 0.3 seconds we discard it, as those are outliers. Thus, we are left with: $MedianHR = 60/(median(IBI_1, IBI_2, IBI_3, IBI_4, ..., IBI_n))$

6.5 Heart Rate Variability

Another useful metric we can extract from the pulse waveform is heart rate variability, or HRV. HRV represent the variance in time between consecutive beats of your heart [14]. To calculate HRV, we simply take the root mean square of the IBI values, $HRV = RMS(IBI_1, IBI_2, IBI_3, IBI_4, ..., IBI_n)$

Chapter 7

Mobile Software for Cardiovascular Disease Screening

7.1 Background and Current Need

In this chapter, we discuss the required mobile software changes needed to support a cardiovascular disease screening solution for deployment in a field test. Over the past few years, our group has developed a series of algorithms and mobile tools to aid in cardiovascular screening. These tools have primarily been used internally, however, for deployment for third-party use, various changes and development was required.

Currently, our Lab supports multiple mobile application packages for various projects including pulmonary, diabetes, and cardio. While work has been done on all three of these disease groups, some functionality required to support third party use had not yet been implemented.

It is also important to always keep the user experience in mind. Because much of our work is with foreign countries, it is important to ensure that the apps are intuitive and minimize the chances for human error despite differences in language barrier. While this may seem trivial, in practice it is often overlooked. For example, some of our machine learning algorithms require data from multiple different measurement apps. Ensuring the user cannot move on to later steps without performing each measurement reduces confusion later on, in the case that the algorithm fails to run because data is missing. This situation can be handled two ways: display an error message at a later step, or prevent the user from moving on without collecting all the data. While both are reasonable options, the former may add more confusion as the reason for error can be unclear. It could be due to the algorithm failing, inaccurate data collected, or that measurements are missing. Thus, it is often preferred to eliminate sources of confusion and error earlier on in the application workflow.

For the scope of this thesis, the PPG NAJA measurement application was left out due to time constraints. Additionally, a new measurement application, PPG Berry, was added to our cardio measurement applications.

7.2 New Mobile Applications

7.2.1 PPG Berry

To meet continued demands for a useful field study, a new application was added to our suite of CVD measurement apps, coined PPG Berry. This application connects to a bluetooth enabled pulse oximeter and obtains a PPG waveform. The intended use is as an alternative to the PPG Camera app. Because both the PPG Camera app and PPG Berry app are used to collect a 30 second recording of a PPG waveform, there is no need to use both applications. Additionally, the recordings from both apps are fed into the PWA algorithms via the `process_measurement` API call.

The PPG Berry app is designed to work with the BM1000C pulse oximeter developed by Shanghai Berry Electronic. It was chosen due to the fact that it is low-cost, and the device is readily available in Bangladesh, which is where this study will be primarily conducted.

The PPG Berry application connects to the BM1000C device via bluetooth, and once a connection is established, the user can run a recording for 30 seconds. In addition to the PPG waveform, we also extract the patients heart rate, and spO2 levels. At the end of the 30 second recording period, the patient can view their waveform and choose to save the measurement or discard the measurement and perform another

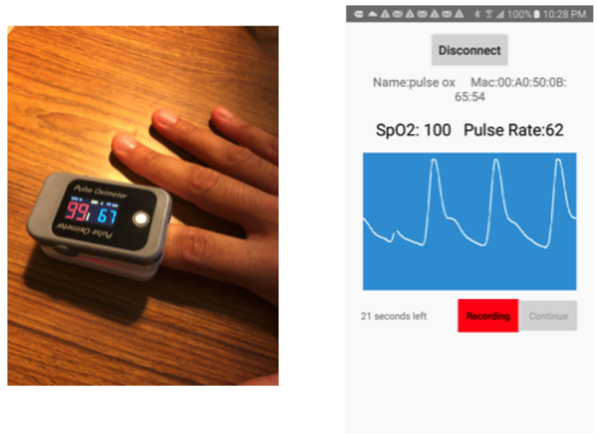


Figure 7-1: Example use of the BM1000C (left) and PPG Berry app recording a measurement (right)

recording. An illustration of this process is shown in 7-1.

Lastly, because the BM1000C device does not transmit a timestamp with each packet, a method for recording the time was devised. The device transmits information at a rate of 100Hz. Upon inspection, a 30 second recording transmits about 2,980 data points. Despite roughly 20 data points being lost in transmission, perhaps due to interference, we use an even interval to represent the time at which each points was received. Each time stamp is recorded in UNIX time (milliseconds). When the a recording is first begun, the time is noted and each data point thereafter is given a value defined by the following formula: $previous\_timestamp + (recording\_start + 30000)/total\_datapoints)$. On average, we should expect to see a data point every 10 milliseconds.

7.3 Stand Alone Measurement Applications

Initially, each of our measurement applications were intended to support both anonymous mode and clinical mode. Clinical mode, however, required the use of the container app in order to maintain a clinician profile, keep track of patient information and study groups. Because of this fact, using the measurement application on it's own in clinical mode was confusing, required additional setup through the container app, and more importantly, led to errors due to misuse. For example, our group had noticed that many people on the Google Play store were downloading our measurement applications and using them incorrectly. This lead to bad reviews, and in some cases, removal of our app from the Play Store.

To mitigate these risks, clinical mode was removed from the measurement apps and the functionality to use these apps standalone was implemented. In order to support this functionality, the required changes were made to each measurement app:

- Decouple clinician and patient registration from taking and analyzing measurements
- Remove authentication and login
- Create a database to support saving and downloading measurements for unregistered patients
- Intuitive visualizations and results

Because most of our applications were intended to be used by clinicians and health workers, authentication of users and privacy of patient information was of great importance. As a result, many measures were taken to ensure that unregistered users could not access our database tables and misuse our applications. With the need to now support anonymous mode, certain changes were necessary to ensure these applications could operate standalone.

7.3.1 Anonymous Mode Database Distinction

On the Android side, because the container app and measurement apps share the same database, it is important to develop a distinguishing factor between whether data was saved into the database anonymously from the container app or from the measurement app. The reason is, all anonymous data from the container app must be deleted with each new patient, however, anonymous data collected from the measurement app when used standalone should be persisted. The solution chosen is simple, when running anonymous mode from the container app we use a patient ID of -1, and when using the measurement app standalone we use a patient ID of 0.

7.4 Cardio Screener Container App Development

For the cardiovascular disease project, the container app is named Cardio Screener. As mentioned above, this app is used by clinicians and health workers to manage patient data, interact with our server as well as measurement applications, and to run our algorithms.

7.4.1 Clinical Mode Development

Patient Diagnosis

Much of the functionality for operational use in the clinical mode had previously been built out by students in our lab, however, there were a few necessary additions. First, the ability for a clinician or health worker to add a patient diagnosis was implemented. The steps to do this are described below, and example illustrations from the cardio screener app are shown in figure 7-2.

- Clinician chooses to add a new diagnosis
- Select measurements that lead the clinician to determine that diagnosis
- Finally, the clinician adds the diagnosis to those measurements

As our cardiovascular disease screening solution collects more data, it is important for clinicians to be able to label that data and provide diagnoses for patients. These labels are used for training our machine learning algorithms, but also can be in a map depicting diseases and their respective locations to different areas. A previous UROP student, Yosef Mihretie, built the initial functionality to display diagnoses on a map for the cardio server, and this is being extended by current UROP student Yogi Sragow.

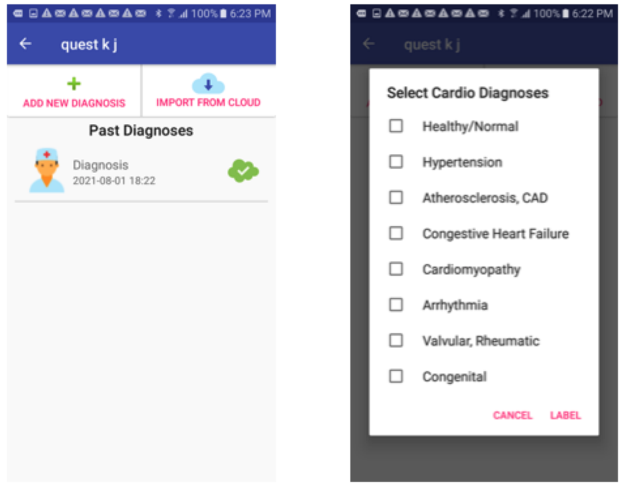


Figure 7-2: Patient's past diagnoses (left) and a clinician adding a new patient diagnosis (right)

Computer Analysis

Another feature that was implemented was a more visual way to display the output of our Bayesian diagnostic and pulse wave analysis (PWA) algorithms. The input to the PWA algorithm is a PPG waveform obtained using our PPG Camera, or PPG Berry (described below), measurement applications. Using this waveform, a previous graduate student, Victoria Ouyang, implemented a function to calculate the average PPG curve. Because this curve is useful to either a clinician, or a patient, in determining the health of that patient's pulse, it was added to a report that is shown as an output of the Bayesian algorithm. Additionally, the parameters that go into calculating the PWA score were included, and color coded based on their standard deviation from the mean so as to make the results more understandable to someone without a former medical training. The way that this is typically done in medicine, is that we compare the parameter to the average in the population. A previous UROP student, Judith Fusman, compiled histograms of our dataset for each of the parameters. Each parameter has three ranges: red, orange, and green. The thresholds used are shown in 7.1. Additionally, in parenthesis we include the Z-score, which is calculated according to the following formula: $Z - Score = (value - mean)/st\_dev$

Parameter (range)	Green	Orange	Red
PWA Score (0-5)	>3	2-3	0-2
Rising Edge Area Ratio (0-1.7)	>0.8	0.6-0.8	0-0.6
SD Parameter (0-0.4)	>0.2	0.1-0.2	0-0.4
Rising Edge Parameter (0-4)	3>	2-3	0-2

Table 7.1: Tests to diagnose cardiovascular diseases [10]

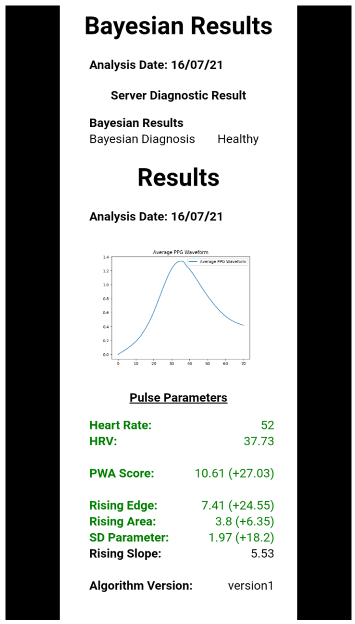


Figure 7-3: Patient diagnostic report generated by the `run_analysis` API call.

Lastly, the report was generated on the server side, and the HTML code was passed back to the android device and displayed in a web view. This is shown in figure 7-3

Process Measurement API Integration

One limitation of our old workflow was that the API endpoint used on the Cardio Server to run the Bayesian algorithm had to handle a very large workload. This endpoint, `run_analysis`, was used to perform any necessary computationally heavy processing on measurements, including signal processing, feature extraction, before then running our machine learning algorithms. Due to the heavy processing being done by the `run_analysis` API, a previous graduate student, Saadiyah Husnoo, broke up this pipeline into two steps:

1. Creation of a new `process_measurement` API which involved any processing or algorithms done on the measurement level

2. Run the diagnostic machine learning algorithm using the `run_analysis` API once all necessary measurements were processed.

The initial behavior was to reduce the workload of our `run_analysis` API and so this pipeline was rewritten on the server side. Meaning, if the Android application called the `run_analysis` API, any unprocessed measurements were analyzed using the `process_measurement` API before the a diagnostic algorithm was run. On the Android application side, the mobile app then simply waited for a response before showing a result.

To further extend the use cases of the `process_measurement` API, each measurement type was now given the ability to process that measurement from within the app, without having to run the Bayesian algorithm. For example, after taking a PPG measurement the user may want to analyze the PPG waveform to gain useful insights without necessarily having to run the Bayesian algorithm.

Other use cases include:

- Directly viewing any graphs or metrics generated by measurement processing algorithms

- Visually and analytically inspecting measurement results before running the diagnostic algorithm

- Identifying errors at the measurement level that may previously have only been identifiable at the diagnostic level

7.4.2 Anonymous Mode Development

To allow users to use our CVD screening app without signing up for the EMR system, anonymous mode support was added to the Cardio Screener. Before we discuss the chosen design and workflow, it is important to define an important use case. One of the use cases of an anonymous mode workflow is for a quick screening of patients in a health camp. For example, in the case where it is necessary to screen many patients in a row, individually signing each patient into the EMR system is time consuming and often unnecessary.

In clinical mode, each patient is given a *home page*, and all further functions are carried out from that central location. For example, from there a clinician can take measurements, view measurements, run an analysis, and so forth. This set-up, however, requires moving back and forth between pages which is inefficient in a setting where you need to perform tests in a more linear fashion.

To address these shortcomings in anonymous mode, a linear workflow was preferred. With this design, there are four key steps:

1. **The clinician performs the measurements, collecting the specified data**

   For screening a patient in anonymous mode, a clinician must take the required measurements. The four measurement options in the cardio screener app are: Cardio Questionnaire, PPG Camera, PPG Berry, and PPG NAJA (this app has been disabled as of the writing of this thesis). To ensure that a clinician has taken measurements before moving on, the continue button is disabled until the measurements have been performed. Additionally, because the PPG Camera and PPG Berry apps both record a PPG waveform, it is not necessary to record both measurements. In the case where both are recorded, only the PPG Berry measurement will be used in the diagnostic algorithm.

2. **The clinician can then view these measurements**

   Once each measurement has been taken, the clinician or health worker can move on to the next screen and view the measurements taken. From here, they can inspect each measurement and ensure that the measurements are of high quality. In the case that they would like to take a measurement again, they can go the previous screen to make another recording. Note, that in order to reduce error going forward, if the clinician decides to take another measurement, the old measurement is deleted.

3. **A diagnostic analysis is run using the selected measurements**

   Once the clinician is satisfied with the quality of the measurements, they can choose to run the diagnostic algorithm. To do this, they select the measurements they would like to include in the algorithm, and then submit these measurements to the server. Upon completion, the server will return a report and display that to the clinician. An example report is shown in figure 7-4. From here, the clinician can choose to run a new analysis, save the report locally to the device as a PNG file, or move on to the next patient.

4. **The clinician can run a new analysis, save the diagnostic report, or move on to the next patient**

   Once the clinician moves on to the next patient, they are taken to the beginning of the anonymous workflow and can carry out the above process again. Additionally, all data collected from the intermediate steps is deleted. This design was chosen for its simplicity, and is meant to simulate a cyclic workflow. While a clinician is interacting with a patient they are moving forward in the workflow, and upon completion they are returned to the beginning to start this process again.

This workflow and images from the application are shown in 7-4.

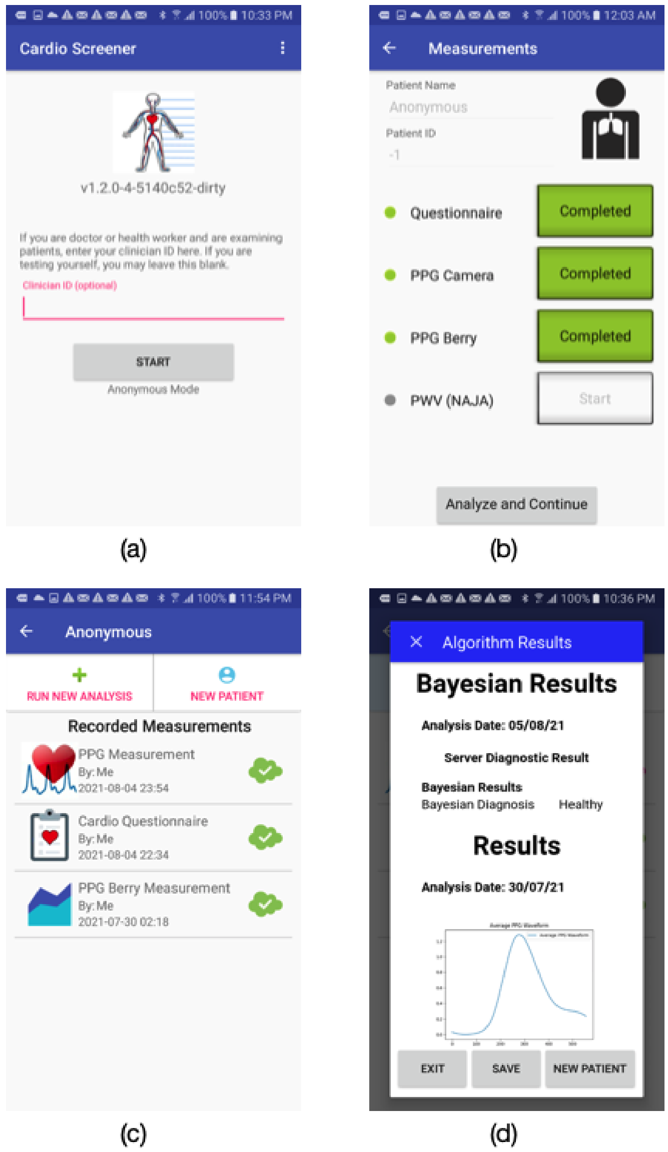


Figure 7-4: Workflow of the cardio screener app in anonymous mode. (a) page to begin screening a new patient, (b) measurement page to record different measurements, (c) view measurements and run diagnostic algorithm, (d) diagnostic report from the bayesian algorithm

7.4.3 API Integration

In the anonymous workflow, only the diagnostic result is saved on our servers, as well as the GPS location of the result. This is used internally to help visualize the prevalence of diseases in different areas. All measurement data is deleted from both the mobile device, and our servers.

Process Measurement Anonymous

In order to run the diagnostic and measurement algorithms in a timely manner, the workflow was split up into various segments. This was designed to alleviate some of the processing done when running the diagnostic algorithm, similar to how it was defined above for clinical mode. However, there are some key differences.

After each measurement was recorded and the clinician moves on to the *View Measurements* page, an automatic processing pipeline is kicked off to analyze each measurement. The Framingham risk score is generated on the mobile device using preset weights on each input question, and does not need to be sent to our servers for further processing. The PPG Camera and PPG Berry measurements, however, must be sent to our servers for further processing before being used in the diagnostic algorithm. Because both the PPG Berry and PPG Camera apps record the a pulse wave form measurement, they both are sent to the same API. To trigger the PWA analysis, a POST request containing a CSV file of the pulse waveform is sent to the `anonymous_diagnostics/analyze_pwa` API. If this response is valid and successful, an ID value is returned. If not, an error is returned. If successful, we persist the ID value and call the the `get_anonymous_results` API to retrieve the result of the analysis. Because some algorithms may take longer than others, we repeatedly call this API until a result is received. Upon completion, results are saved locally on the device. An example result is shown in 7-5:

Note here, that the `file_payload` field is unique to the PWA Analysis pipeline, and contains the average PPG curve. This is then done for each measurement.

After each measurement has been analyzed, the user can then run the diagnos-

```
{
    "created_at": "1617130000",
    "result": {
        "algorithm_version": 'version1',
        "pwa_score": 3.74,
        [ ... other fields]
    },
"file_payload": "/uploads/AnonymousAPIResult/[
    average_curve_file_path]",
"error": null
}
```

Figure 7-5: An example anonymous PWA response

```
bayesian_data = {
    "analysis_type": "bayesian",

    #from questionnaire
    "weight_kg": 75,
    "height_cm": 178,

    #from pwv result
    "PWV_ET": 15.0,
    "PWV_EF": 15.0,

    #from PWA result
    "pwa_score": 4.8,

    #location
    "gps_latitude": 123.45,
    "gps_longitude": 123.45

}
```

Figure 7-6: An example JSON request

tic algorithm. For this to happen, a JSON object is constructed from the results of the individual `process_measurement` calls, as shown in 7-6. This is then sent to the `anonymous_diagnostics_analyze_bayesian/` API. Similarly as described above, the response is either an ID or an error. If successful, the ID sent to the `get_anonymous_analysis_result` API. This endpoint is repeatedly queried until the diagnostic report is received, and this is displayed to the user.

It is important to note the difference here between the clinical and anonymous workflows. In clinical mode, measurements are saved on the server so it suffices to simply send each measurements ID to to the diagnostic algorithm. From there, the server can look up the relevant measurements and extract the necessary data. In anonymous mode, however, the measurements are not saved on the server for an indefinite period of time, and so it is necessary to instead format the required request on the user end.

7.5 Android Database Schema

To maintain consistency across our apps, we chose to implement a database structure that would be simple to replicate across different applications. All of our apps store information in a home directory named *MobileTechLab*. Within this folder, there is a folder for each container application (Cardio, Pulmonary, Diabetes, etc.), and additionally for each measurement application (PPG_Berry, PPG_Camera, Cardio_Questionnaire). The container application contain the database tables for the patients, clinicians, and measurements associated with that container app. Additionally, there is also a folder *AnonymousResults* where the report from the anonymous diagnostic algorithm is stored.

The measurement applications store larger data associated with each measurement. For example, in the case of the PPG_Berry and PPG_Camera apps this may include the CSV file of the PPG waveform. These folders are further broken down into an *Anonymous* and *Clinical* Folder. The *Clinical* Folder contains a folder for each patient, under their patient ID number, with the measurement data. The *Anonymous* folder is further broken down into two folders, a *temp* folder and *Measurements* folder. The *temp* folder contains temporary measurements collected when screening patients in anonymous mode from the screener application. This data is deleted every time a clinician moves on to a new patient. On the other hand, the *Measurements* folder contains measurement data collected when using the measurement app standalone. This data is not deleted, unless the user chooses to delete the data. This hierarchy, with emphasis on the PPG Berry measurement app, is illustrated in 7-7

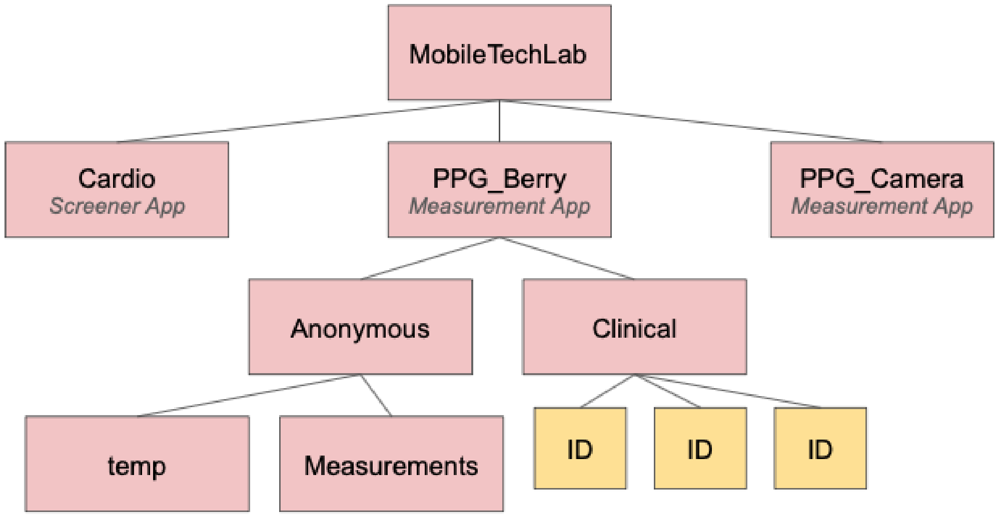


Figure 7-7: Android database hierarchy, with a focus on the PPG Berry measurement app

7.6 Android Updates

In addition to the Android updates described above, each Android app used in the Cardio project was updated and compiled to the latest version of Android, and is using the most recent packages and libraries as of this writing.

Chapter 8

Server Development for Cardiovascular Disease Screening

8.1 Additional API Changes

The cardio server is intended to work with our cardiovascular disease screening applications. This includes, Cardio Screener, PPG Camera, Cardio Questionnaire, PPG NAJA, and more recently PPG Berry. The main client to this server is the Android applications.

As described in Chapter 7, the cardio server has been recently updated by a previous graduate student, Saadiyah Husnoo, to support new functionality necessary for deployment in a field test. These updates include the addition of an anonymous mode, as well as breaking up the diagnostic algorithm workload, among many other additions [15].

8.2 Berry Pulse Oximeter

With the addition of the PPG Berry measurement application to our suite of cardiovascular diseases screening tools, it was necessary to make the proper adjustments on the server code to be able to handle any incoming requests from the PPG Berry app. More specifically, this involved three main changes: support to add PPG

Berry measurements to our server database, process those measurements using the `process_measurement` API, and lastly, updated support in the `run_analysis` API so that these measurements can be properly utilized in the diagnostic algorithm. Additionally, this functionality had to be added for both our anonymous and clinical APIs.

When a measurement is uploaded to the server, it is stored in the database as a model containing passed fields from the measurement. For the PPG Berry application, the database fields include a CSV file of the pulse waveform, as well as the heart rate and spO2 values calculated using the BM1000C device. Additionally, a metadata model called `DiagnosticMeasurementMetaData` is also created and saved to the database. This model contains information including the measurement, as well as other metadata such as recording time, which patient the measurement belongs to, and the GPS coordinates of the measurement.

8.2.1 Add Measurement Support for Clinical Mode

The Add Measurement API is called by the mobile applications in the form of a POST request. The request is made when the client would like to upload the measurement to be saved on the server, and includes the relevant data in the form of a JSON object. In order to add a PPG Berry measurement to the server, a new endpoint at the URL `add_measurement/pwa_berry/clinician/` was created. An example formatted post request is shown in 8-1.

Once the request is received, the request serializer converts the parsed data into a Django object. The model is then validated to ensure that the request has all the necessary fields. At this step, it is important that both the server and Android app keep the field names consistent. Failure to do so will result in errors, and the model will fail to validate the request. If successful, the request will return a response containing the ID of that measurement which can be looked up at a later time, or used in subsequent API calls.

Once the measurement has been added to the server, the necessary changes to ensure the `process_measurement` and `run_analysis` API calls work as intended

```
{
    "usergroup_id": self._group.identity,
    "patient_id": self._patient.identity,
    "measurement": {
        "metadata": {
            "recorded_on": "2018-04-21T02:00",
            "duration": 60,  # seconds.
            "client_id": "BTester",
            "client_type": CLIENT_TYPE.WEB_APP.code,
            "client_version": "101",
            "gps_longitude": 123.1234567,
            "gps_latitude": 123.1234567,
        },
        "pwa_berry_measurement": {
            "pulse_time_series": self.generate_sample_upload_file(),
            "pulse_rate": 67,
            "spo2": 99
        },
    },
}
```

Figure 8-1: An example PPG Berry add measurement request

is to simply add the correct support for how the code should react in the case of a PPG berry measurement. Because the PWA analysis pipeline is called for both the PPG Camera measurements and PPG Berry measurements, minimal changes were made. An example of the handling of PPG Berry measurements in a call to `process_measurement` is shown in figure 8-2

8.2.2 Anonymous API Changes

In anonymous mode, we don't save measurements and as a result, there is no `add_measurement` API call. Instead, we simply pass the pulse waveform obtained from the PPG Berry app to the `anonymous_diagnostics/analyze_pwa`. We then parse the response for the measurement ID, and use that value for subsequent calls to obtain the results from the analysis, as described in the previous chapter.

```
##################### handle PWA Berry measurements
    ##########################

#all pwa berry measurements (DiagnosticMeasurement)
pwa_berry_measurements = allowed_measurements.filter(
    pwa_berry_measurement__isnull=False)

#all pwa berry measurements that already have an associated PWAResult
#(DiagnosticMeasurement), then we get associated PWAResult
current_pwa_version = default_pwa_classifier.
    get_current_algorithm_version()
analyzed_pwa_berry_measurements = pwa_berry_measurements.filter(
    identity__in=PWAResult.objects.filter(algorithm_version=
    current_pwa_version).values(
        "measurement"))

#Unanalyzed pwa measurements that don't have an associated
#PWAResult (DiagnosticMeasurement)
unanalyzed_pwa_berry_measurements = pwa_berry_measurements.exclude(
    identity__in=analyzed_pwa_berry_measurements)

# Analyzing each of these unanalyzed pwa measurements
for pwa_berry_measurement in unanalyzed_pwa_berry_measurements:
    transaction.on_commit(
        lambda: get_pwa_berry_diagnostic.delay(pwa_berry_measurement.
    identity, None)
    )
```

Figure 8-2: PPG Berry process measurement request handling

Chapter 9

Pulmonary Disease Screening

9.1 Pulmonary Disease

Pulmonary diseases represent another major category of diseases that account for a large proportion of global deaths. Pulmonary diseases affect the lungs and other parts of the respiratory system, and include asthma, chronic obstructive pulmonary disease (COPD), pneumonia and pulmonary fibrosis to name a few. Pulmonary diseases can be caused by a multitude of factors, but some of the more common factors include exposure to noxious agents present in the air, pollution, biomass cooking, and both primary and secondary tobacco smoke. While these factors affect both developed and under-developed nations, their prevalence in underdeveloped nations is especially high due to early childhood disadvantages and constant exposure to indoor and outdoor air pollution.

9.2 Diagnosing and Screening Pulmonary Disease

Similarly to cardiovascular disease diagnoses, when diagnosing and screening for pulmonary diseases the standard procedure consists of an assessment of current symptoms, family history and clinical history, as well a series of pulmonary function tests described in 9.1.[16]. Pulmonary function tests measure the amount of air a patient can inhale and exhale, and whether or not enough oxygen is delivered to the blood.

Test	Description
Spirometry	During this test, patients blow into a spirometer which measures FEV1, the forced expiratory volume of a patient in 1 second, as well as FVC, the forced vital capacity, or total amount exhaled
Body Plethysmography	The patient inhales and exhales into a breathing tube while sitting inside an airtight box. The pressure in the box is adjusted in order to determine the absolute volume of air in the lungs.
Diffusing Capacity	This test measures the ability of the lungs to transfer oxygen from the alveoli into the blood stream

Table 9.1: Pulmonary function tests to diagnose pulmonary diseases [16]

Other tests inlude a 6 minute walk test, pulse oximetry, as well as CT scans and X-rays. In the developing world, access to these tests and tools is often not available, and so medical care is generally centered around questionnaires and basic readings obtained from a stethoscope.

9.3 Challenges of Detecting Pulmonary Disease

In developing nations, there is a rising trend in the prevalence of pulmonary diseases. [17] This is due to increased tobacco use, environmental pollution, biomass cooking, and poor air quality. Additionally, diseases like asthma and COPD are chronic conditions that affect quality of life in the long term, it becomes exceedingly important to identify at risk individuals early on before symptoms progress to more advanced stages. This could drastically improve quality of life.

The challenge in diagnosing and screening for pulmonary diseases in low-resource settings lay with the fact that much of the tools used to perform accurate assessments are expensive, and not readily available. As a result, screening is generally conducted using simple questionnaires and other easily accessible tools. While these can be effective, without pairing them with other measurements they can prove misleading. For example, it is believed that the prevalence of smoking, air pollution and environmental hazards are considerably higher in developing countries, however, as high as 30-40% of Asian lunch cancer patients had never smoked, compared to 10% of patients in the US. [17]

9.4 Our Approach to Pulmonary Disease Screening

Similarly to our approach for cardiovascular disease screening, our group has developed a holistic mobile platform with several measurement applications, including a general pulmonary questionnaire, stethoscope, peak flow meter, thermal screener, 6-minute walk test, pulmonary function test, and spirometry test. Each of these measurements can be performed using either a mobile smartphone, or using a provided auxiliary device, and are collected using our mobile applications.

A previous graduate student, Daniel Chamberlain, developed the initial mobile toolkit used to screen for pulmonary disease, shown in Figure 9-1. The thermal camera, which was later added to this toolkit, is absent from this image.

The measurement applications used in our tooklit are described below [18]:

- **Questionnaire**: The general pulmonary questionnaire includes a series of questions regarding medical history (the onset, duration, and progression of breathlessness, coughing, nasal symptoms, chest pain, fever) , family history, basic questions (sex, weight, age, etc.) as well as questions pertaining to risk factors (alcohol usage, smoking, etc.) . This questionnaire was initially developed to screen for COPD and asthma, but was updated to also include questions pertaining to COVID-19.

- **Peak Flow Meter**: The peak flow meter is a low-cost, easy transportable tool used to determine lung performance. To use the device, patients blow quickly into the mouthpiece. The force of air pushes a marker back along a numbered scale, and the point at which it stops is the peak flow rate, in liters per minute. A mobile application was developed that uses virtual reality and the phones camera to locate record a series of measurements and save the patients peak flow rate.

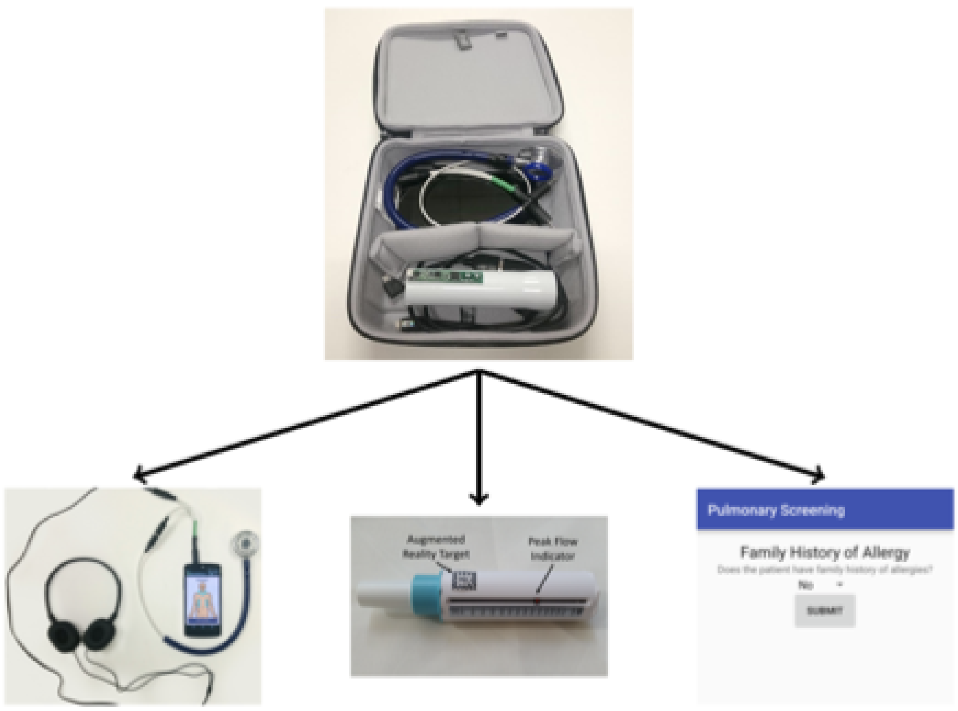


Figure 9-1: A mobile diagnostic toolkit used for pulmonary screening [18]

Figure 9-2: Thermal camera used in the mobile diagnostic toolkit (left) and the camera in use on a patient at the Chest Research Foundation (right) [18]

- **Electronic Stethoscope**: A custom stethoscope was designed by our lab to record lung sounds using an Android device. Lung sounds are recorded from eleven different sites on the body, and cough sounds are recorded from the trachea.

- **Thermal Camera**: A thermal camera connects to the smartphone via the micro-USB port, and is used to collect thermal images of the patient's chest, side, and back during inhalation and exhalation. The thermal camera is shown in figure 9-2

- **Pulmonary Function Test**: This test is used to record all the relevant information from a pulmonary function test inlcuding information from a spirometry test, body plethysmography, DLCO gas diffusion, as well as impulse oscillometry.

- **Other Tests**: Additional tests include a spirometry test which is a questionnaire that records the answers to a standard spirometry test. A 6-minute walk test measurement app is also used to record and capture the results to a standard walk test.

Resulting data is stored in our EMR system that stores and manages patient information, and is used in our Bayesian diagnostic algorithm to return a diagnosis report to the user.

Chapter 10

Pulmonary Disease Algorithms and Machine Learning

As done in the cardiovascular project, in pulmonary we similarly model disease diagnosis using a Bayesian network. The network, developed by graduate student, Aneesh Anand, uses a 3-layer Bayesian network [18]. Details of the model are further explained here.

10.1 Network Design

The Bayesian network structure used to diagnose pulmonary diseases is represented in three layers: risk factors, diseases, and symptoms. Risk factors are activities that puts an individual at risk of contracting a disease. This may include activities such as smoking, inhalation of biomass fuels, heavy use of alcohol, etc. The disease layer represents the absence or presence of each individual disease we seek to predict, and the symptoms layer represents the features that result from having a particular disease. For example, the symptoms layer may contain presence of cough, fever, nasal congestion, etc. A simple 3-layer network is shown in figure 10-1.

Once the nodes and features were determined, the model needed to be fully parametrized. First, to establish the prior probabilities, or the likelihood of variables that don't have parents, data from population-level studies was used. In the

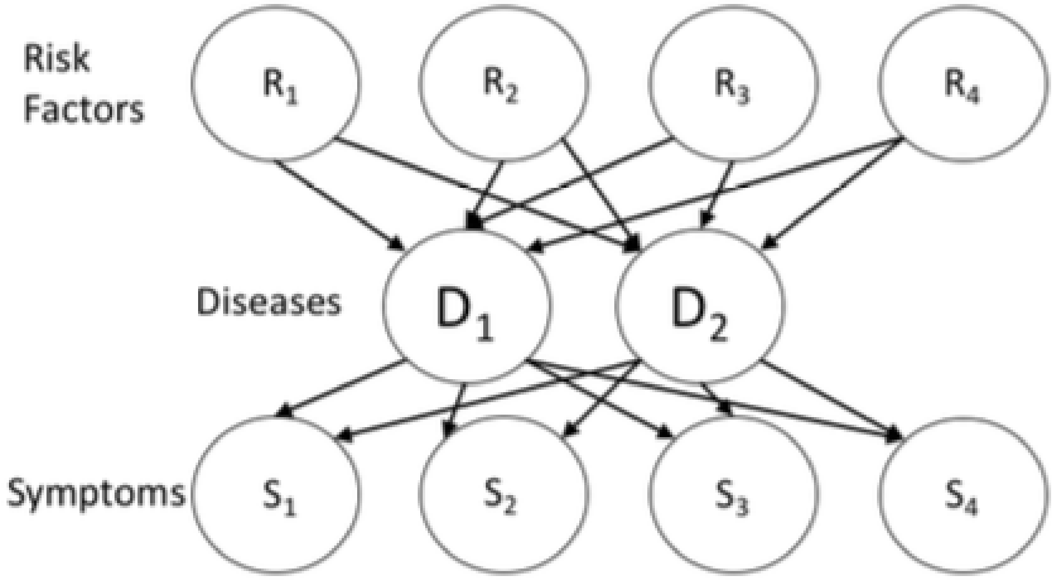


Figure 10-1: A simplified 3-layer Bayes network used for disease diagnosis [18]

case of our model, these variables are the risk factors.

Next, the conditional probabilities needed to be determined. These probabilities signify the strength of the edges connecting the variables in the model. In the current model being used on the pulmonary server, these parameters were derived from previously collected data.

10.2 Pulmonary Bayesian Network

The pulmonary model currently predicts the following diseases: COPD, Asthma, Allergic Rhinitis, or Other (this group includes ILD, pneumonia, and TB). The risk factors, composed of the characteristics that can give a patient a higher risk for pulmonary disease, were based off of the results of a questionnaire. The symptoms were taken from both the questionnaire and the mobile took-kit developed by our lab. A diagram of the network architecture is shown in figure 10-2

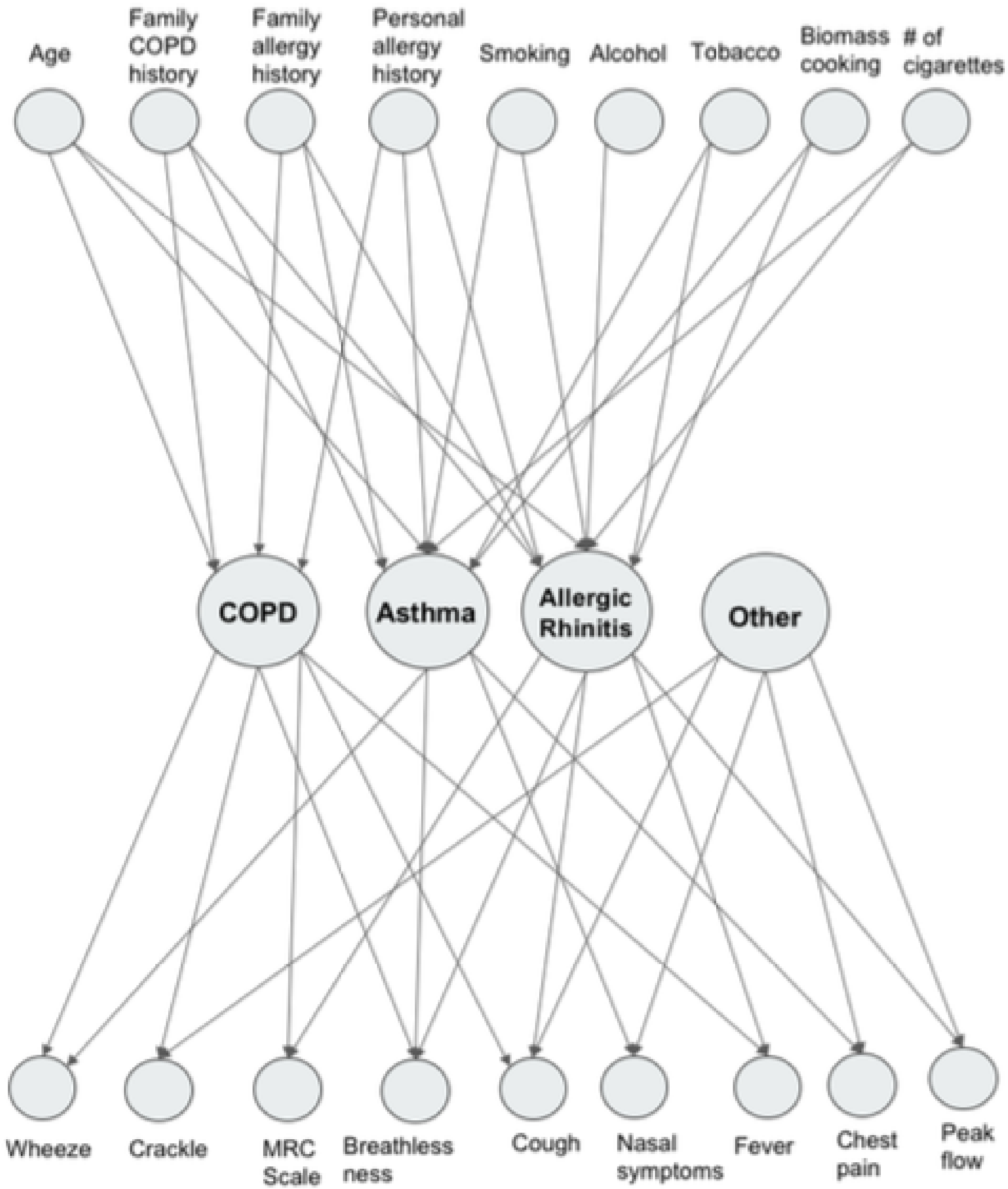


Figure 10-2: Bayesian network used for disease diagnosis. Some arrows have been omitted for clarity [18]

Chapter 11

Mobile Software for Pulmonary Disease Screening

11.1 Introduction

In the previous sections, we discussed the diagnostic algorithm used to generate a diagnosis, as well as traditional methods for pulmonary diagnosis and screening. Here, we describe the mobile implementation of our diagnostic tool, coined Pulmonary Screener, and the changes made to support ongoing and continued field studies. The pulmonary tools will be employed for a new clinical study in Dhaka, Bangladesh. Our clinical partner for this study is the National Institute of Diseases of the Chest and Hospital (NIDCH).

11.2 Mobile Application Relationship

Similarly to what was described in Chapter 7, the pulmonary project utilizes and container application, and a series of measurement applications. The container application, Pulmonary Screener, is used as an interface to our EMR system and to interact with the algorithms on our server.

Currently, there is no anonymous mode for either the pulmonary measurement apps, or the container app. In future iterations of our mobile screening tools, these

COVID-19 Questions
1. Are you feeling tired, fatigue, or weakness?
2. How long have you been felling tiredness and fatigue?
3. Have you had a sudden loss of smell or taste?
4. For how long have you had a loss of smell/taste?
5. Has patient experienced any recent nausea or diarrhea?
6. How long have you had nausea or diarrhea?
7. Has patient experienced any recent rash on skin?
8. How long have you had a rash on your skin?
9. Have you been experiencing any confusion of trouble thinking?
10. How long have you been experiencing confusion or trouble thinking?
11. Have you travelled recently (in the past 10 days)?

Table 11.1: COVID-19 questions added to the pulmonary questionnaire measurement app. Note, follow up questions are only asked if the patient answered 'yes' to the previous question

features will be implemented.

11.3 COVID-19 Support

During the COVID-19 pandemic, there was an increased need to be able to quickly and efficiently screen patients to determine if they needed to take extra precautions. Additionally, as the disease was not well understood collecting data was crucial to understanding its progression and developing further treatment plans. To meet this demand, we added a series of questions to our general pulmonary questionnaire. The questions are outlined in table 11.1

Additionally, we added the ability for clinicians to input a COVID-19 diagnostic label into our EMR system. This is currently being used by our partners in India. As we collect and gather more data, hopefully this will be used to train a COVID-19 diagnostic model and can be deployed as a screening tool in various healthcare

Pulmonary Blood Test Questionnaire
1. Total Leucocyte Count (WBC)
2. Neutrophil Percentage
3. Lymphocyte Percentage
4. Eosinophil Percentage
5. Monocyte Percentage
6. Blood Platelet Count

Table 11.2: Questions asked in the blood test questionnaire app

settings.

11.4 Blood Test Questionnaire

One of our partner doctors in India discovered that blood tests may be an early indicator of COVID-19 before symptoms appeared. If this was indeed the case, it was important to collect blood test information on patients in the area. To do this, there were two apparent approaches: we could add these questions to the current pulmonary questionnaire, or create a new measurement app.

Currently, the pulmonary questionnaire is being used as a screening tool for multiple different pulmonary diseases. It is designed to be used as a general screening tool, and does not require any equipment or access to a lab. Thus, it was decided to leave the pulmonary questionnaire as is, and instead create another measurement application for use by doctors who do have laboratory access. This app is called Blood Test Pulmonary Questionnaire, and is intended to collect information from blood tests. The questions included are shown in table 11.2

Chapter 12

Server Development for Pulmonary Disease Screening

In this chapter we describe the pulmonary server, intended to work with our pulmonary disease screening applications. The main client to this server is the Android applications, and the architecture is very similar to that of the cardio server described previously. The server platform is crucial for the integration and scalability of our screening tools.

12.1 Additional API Changes

With the addition of the Pulmonary Blood Test Questionnaire App to our suite of pulmonary disease screening tools, it was necessary to add support on the server code to be able to handle any requests from the Blood Test Android application.

Maintaining this data on our servers is crucial for the ongoing COVID-19 study. As more data is uploaded from our partner doctors in India, our group will be able to download this data and use it to analyze the prevalence of COVID-19 and potentially detect any early indicators of the disease. Additionally, with this training data we can work on models to further screen for and diagnose COVID-19.

```
{
            "usergroup_id": self._group.identity,
            "patient_id": self._patient.identity,
            "measurement": {
                "metadata": {
                    "recorded_on": "2018-04-21T02:00",
                    "duration": 60,  # seconds.
                    "client_id": "BTester",
                    "client_type": CLIENT_TYPE.WEB_APP.code,
                    "client_version": "101",
                    "gps_longitude": 123.1234567,
                    "gps_latitude": 123.1234567,
                },
                "pulmonary_blood_test_questionnaire": {
                    "leucocyte": 50.0,
                    "neutrophil": 50.0,
                    "lymphocyte": 50.0,
                    "eosinophil": 50.0,
                    "monocyte": 50.0,
                    "blood_platelet_count": 50.0
                },
            },
        }
```

Figure 12-1: An example Pulmonary Blood Test Questionnaire add_measurement request

12.1.1 Blood Test Questionnaire

After the user records a questionnaire using the mobile phone, the results are stored locally on the device until a connection to our remote servers is established. At that point, the user has the option of uploading data to the cloud.

Measurements uploaded to the server are stored as models in the database. For the questionnaire, the database fields contain the answers to the questions asked in the questionnaire. To support the new blood test questionnaire measurement, a new API endpoint at the URL `add_measurement/blood_test_questionnaire/clinician/` was created. The Android application sends an HTTP POST request with the answers to the questionnaire, where upon receiving the request, the request serializer converts the parsed data into a Django object. The model is then validated to ensure all of the required fields were present in the request, and if so, a DiagnosticMeasurement object is saved in the database. An example formatted POST request is shown in 12-1

12.2 Future Work

As of this writing, the pulmonary Android apps can only operate in clinical mode. In order to use anonymous mode within the Android applications, similar to that of cardio server, new anonymous APIs will need to be implemented.

Chapter 13

Scaling and Deployment of Health Diagnostic Platforms

13.1 Motivation

Our servers play a very important part in health diagnostics. Not only do they hold all of our patient data, but they also contain our algorithms used for measurement processing and diagnosing diseases. Internally, the PyMed EMR infrastructure has been used for conducting studies, maintaining patient information and data, and allowing for the monitoring and diagnosing of patient health. While it has been instrumental for our personal use, our Lab has never deployed our system for outside project use and third-party partners.

In many cases, global health organizations may be constrained to collect and store their own data without passing that information either outside the country, or to other organizations. This limits exactly how much technology these organizations can use, and unless they build their own algorithms and models, they may never be able to realize the full potential of computer-aided health diagnostic systems. A system that would allow these groups to process their data using third party algorithms stored in the cloud, without having them need develop their own algorithms, would be very beneficial.

Here, we describe the various additions that are needed in order to deploy our platform for other organizations to use.

13.2 Addressing Server Requirements

13.2.1 Server Replication and Deployment

Due to governmental constraints, many health care groups are not permitted to send patient data outside of the country. This is a problem if they would like to use our EMR system, as all data is stored on our servers located in the United States.

A major step in allowing other groups to use our EMR system, is the to first clone our a version of our server code and deploy it on another server located in the host country. The steps to do so are outlined below:

- Clone a copy of the server code
- Remove all proprietary algorithms
- Set the new remote git url
- Delete the git history

Note, it is important to be very careful with these steps to ensure that others will not be able to use the git history to revert any changes and access previous code. Additionally, extra attention must be paid to the submodules used in the repository.

After the server code has been duplicated onto the third-party server, they should be to interact with our EMR system and maintain patient data. Note, because we removed our proprietary algorithms, this new server is designed only to keep track of and maintain patient data. It cannot be used to run our measurement and diagnostic algorithms.

13.2.2 Adding Anonymous Mode

For groups that would like to use our algorithms, but are not permitted to use our EMR system, we have developed an Anonymous API that allows users to download our apps, and anonymously use the labs measurement and diagnostic algorithms.

While all patient data is removed from our servers, we do keep anonymized diagnostic result data to be used internally by our lab for visualizing the location and prevalance of different diseases. This data is stored in the `AnonymousAnalysisResult` table and includes GPS longitude, GPS latitude, and the diagnostic result generated by our diagnostic algorithms.

Currently, this functionality is only available on the Cardio Server.

13.3 Addressing Mobile Client Requirements

13.3.1 Supporting Arbitrary IP Address

Lastly, to support use of our apps with custom servers, we need the ability to custom set the IP address within the Android applications. This is done using the mobile app, and a text-based input.

13.4 India COVID-19 Study

As an initial test of our updated PyMed platform for pulmonary diseases screening, a clinical study is currently underway with our existing partners in Pune, India (Dr. Dhadge and a government clinic). The study is collecting data used to develop and train new algorithms for predicting COVID-19, as well as other infectious pulmonary diseases. This ongoing test features the expanded Pulmonary Questionnaire mentioned in Chapter 11, as well as the new pulmonary blood test questionnaire. Data collection using these new tools has been going well, and results are expected to be published in 2022.

13.5 Field Test Implementation in Bangladesh

13.5.1 General Overview

Currently, our group is working with a sponsor in Bangladesh, Smart Innovations Pvt Ltd, on two ongoing studies based in Dhaka. They are using our current server infrastructure for a field test in Bangladesh.

13.5.2 Community Health Worker Study

In order to test the use of our disease screening tools in a rural low resource setting, we have designed a study to evaluate the feasibility and efficacy of these tools in the hands of community health workers. The two sites chosen for this study include the low-income Chittagong region and Jamalpur. For this study, health workers will be going door to door and doing general health screening. Because patients will not need to be registered in our system, we will be employing anonymous mode.

For evaluation, we are recording the location and result of all diagnostic tests and comparing the prevalence of cardiovascular and pulmonary disease against the internal health ministry records of these regions. This study will also be evaluating the feasibility for low-skilled community health workers to use our mobile health tools.

13.5.3 Bangladesh COVID-19 Study

To further test our updated pulmonary diagnostic platform and to collect our initial data for developing a COVID-19 prediction model, we are currently starting a separate study using the pulmonary server and related mobile apps to collect data from coronavirus patients at a hospital in Bangladesh. Our clinical partner for this study is the National Institute of Diseases of the Chest and Hospital (NIDCH). For this study, we are using the expanded pulmonary questionnaire as well as the thermal image camera. The data collected will be combined with the data obtained from the India COVID-19 study to further develop and validate a machine learning model for COVID-19.

Chapter 14

Summary and Conclusions

Building on prior work in our group, the work in this thesis presents the additional requirements needed for actual deployment of server platforms in a real world environment. As mentioned previously, this additional software development includes:

- **Anonymous version of mobile apps**

  In order to support ongoing clinical studies while adhering to local regulation our mobile applications needed to support an anonymous functionality. Beyond this, an anonymous mode also allows groups to utilize our algorithms without having to register with our EMR system. In this thesis, we described a new anonymous workflow for the rapid screening and diagnosis of patients.

- **Anonymous API server integration**

  Previus student, Saadiyah Husnoo, developed an Anonymous API to work with our cardio server. To realize the full potential of the new API, it was integrated into our suite of CVD screening mobile applications.

- **Measurement App Standalone Functionality**

  Each of our measurement apps in the CVD screening project were modified to support offline and independent use from the container apps. This was done to offer meaningful statistics to people that want to record various measurements.

- **New Mobile Applications**

  To keep up with current needs, we described two new apps created for the cardio project and pulmonary project: the PPG Berry app and Blood Test Questionnaire Apps. These apps are being used in an ongoing study in Bangladesh and India, and will further contribute to the development of our diagnostic algorithms and predictive models.

- **COVID-19 Support**

  Due to the current pandemic, there was a need to collect data on patients with COVID-19 to further develop new machine learning models. To support this, we added a new set of questions to our general pulmonary questionnaire, added clinician label support, as well as develop a new mobile application for recording the results of a blood test, which may indicate early signs of COVID-19.

Lastly, there are currently several ongoing clinical studies that are evaluating this server platform in both India and Bangladesh. We are hopeful that these tools and this additional field testing will provide the opportunity to deploy these diagnostic platforms more widely.

Bibliography

[1] Amit Bhondve, B. Pathak, and Rukman M. Manapurath. Mixed-method analysis of community health camps: A novel approach beckoning. *Indian Journal of Community Medicine : Official Publication of Indian Association of Preventive & Social Medicine*, 44:233 – 237, 2019.

[2] Charu Kohli, Jugal Kishore, Shantanu Sharma, and Harsavsardhan Nayak. Knowledge and practice of accredited social health activists for maternal healthcare delivery in delhi. *Journal of family medicine and primary care*, 4:359–63, 07 2015. doi:10.4103/2249-4863.161317.

[3] Smisha Agarwal, Siân Curtis, Gustavo Angeles, Ilene Speizer, Kavita Singh, and James Thomas. The impact of india's accredited social health activist (asha) program on the utilization of maternity services: a nationally representative longitudinal modelling study. *Human Resources for Health*, 17, 08 2019. doi:10.1186/s12960-019-0402-4.

[4] Brian Wahl, Aline Cossy-Gantner, Stefan Germann, and Nina R Schwalbe. Artificial intelligence (ai) and global health: how can ai contribute to health in resource-poor settings? *BMJ Global Health*, 3(4), 2018. URL: https://gh.bmj.com/content/3/4/e000798, arXiv:https://gh.bmj.com/content/3/4/e000798.full.pdf, doi:10.1136/bmjgh-2018-000798.

[5] Youfa Wang, Hong Xue, Yaqi Huang, Lili Huang, and Dongsong Zhang. A systematic review of application and effectiveness of mhealth interventions for obesity and diabetes treatment and self-management. *Advances in Nutrition:*

An International Review Journal, 8:449–462, 05 2017. doi:10.3945/an.116.014100.

[6] Public health programmes that deliver impact at scale. https://www.alemhealth.com/global-health-solutions. Accessed: 2021-7-29.

[7] Openmrs. https://openmrs.org/. Accessed: 2021-7-29.

[8] D-tree. https://www.d-tree.org/child. Accessed: 2021-7-29.

[9] Pietro Modesti, Piergiuseppe Agostoni, Agyemang Charles, Sanjay Basu, Athanase Benetos, Francesco Cappuccio, Antonio Ceriello, Stefano Del Prato, Robert Kalyesubula, Eoin O'Brien, Michael Kilama, Stefano Perlini, Eugenio Picano, Gianpaolo Reboldi, Giuseppe Remuzzi, David Stuckler, Marc Twagirumukiza, Luc Bortel, Ghassan Watfa, and Gianfranco Parati. Cardiovascular risk assessment in low-resource settings: A consensus document of the european society of hypertension working group on hypertension and cardiovascular risk in low resource settings. *Journal of hypertension*, 32, 02 2014. doi:10.1097/HJH.0000000000000125.

[10] Mayo clinic. heart disease - diagnosis and treatment.

[11] Shuchi Anand, Christina Bradshaw, and Prabhakaran Dorairaj. Prevention and management of cvd in lmics: Why do ethnicity, culture, and context matter? *BMC Medicine*, 18:7, 12 2020. doi:10.1186/s12916-019-1480-9.

[12] Peter W. F. Wilson, Ralph B. D'Agostino, Daniel Levy, Albert M. Belanger, Halit Silbershatz, and William B. Kannel. Prediction of coronary heart disease using risk factor categories. *Circulation*, 97(18):1837–1847, 1998. doi:10.1161/01.CIR.97.18.1837.

[13] Michel Ducher, Emilie Kalbacher, François Combarnous, Jerome Vilaine, Brigitte McGregor, Denis Fouque, and Jean Fauvel. Comparison of a bayesian network with a logistic regression model to forecast iga nephropathy. *BioMed research international*, 2013:686150, 11 2013. doi:10.1155/2013/686150.

[14] Task Force of the European Society of Cardiology the North American Society of Pacing Electrophysiology. Heart rate variability. *Circulation*, 93(5):1043–1065, 1996. doi:10.1161/01.CIR.93.5.1043.

[15] Saadyah B. Husnoo. A scalable server platform and api design for real-time health monitoring and diagnostics. *Masters Thesis, MIT 2021.*

[16] Mayo clinic. copd - diagnosis and treatment.

[17] Abhishek Shankar, Deepak Saini, Anusha Dubey, Shubham Roy, Sachidanand Jee Bharati, Navneet Singh, Meghal Khanna, Chandra Prakash Prasad, Mayank Singh, Sunil Kumar, Bhawna Sirohi, Tulika Seth, Minakshi Rinki, Anant Mohan, Randeep Guleria, and Goura Kishor Rath. Feasibility of lung cancer screening in developing countries: challenges, opportunities and way forward. *Translational Lung Cancer Research*, 0(0), 2019. URL: https://tlcr.amegroups.com/article/view/27776.

[18] Aneesh Anand. Bayesian models for screening and diagnosis of pulmonary disease. *Masters Thesis, MIT 2018.*

[19] Amit Levy, Bradford Campbell, Branden Ghena, Pat Pannuto, Prabal Dutta, and Philip Levis. The case for writing a kernel in rust. *Proceedings of the 8th Asia-Pacific Workshop on Systems - APSys17*, 2017. doi:10.1145/3124680.3124717.

[20] Dorothy Lall, Nora Engel, Narayanan Devadasan, Klasien Horstman, and Bart Criel. Models of care for chronic conditions in low/middle-income countries: a 'best fit' framework synthesis. *BMJ Global Health*, 3(6), 2018. URL: https://gh.bmj.com/content/3/6/e001077, arXiv:https://gh.bmj.com/content/3/6/e001077.full.pdf, doi:10.1136/bmjgh-2018-001077.

[21] Rachel Nugent. Chronic diseases in developing countries health and economic burdens. *Annals of the New York Academy of Sciences*, 1136:70–9, 02 2008. doi:10.1196/annals.1425.027.

[22] M. R. Miller, R. Crapo, J. Hankinson, V. Brusasco, F. Burgos, R. Casaburi, A. Coates, P. Enright, C. P. M van der Grinten, P. Gustafsson, R. Jensen, D. C. Johnson, N. MacIntyre, R. McKay, D. Navajas, O. F. Pedersen, R. Pellegrino, G. Viegi, and J. Wanger. General considerations for lung function testing. *European Respiratory Journal*, 26(1):153–161, 2005. URL: `https://erj.ersjournals.com/content/26/1/153`, `arXiv:https://erj.ersjournals.com/content/26/1/153.full.pdf`, `doi:10.1183/09031936.05.00034505`.