Smartphone COVID Infection Risk Assessment

A Major Qualifying Project Report

Submitted by:

CAITLIN ENRIGHT

THEODOROS KONSTANTOPOULOS

ORLANDO PINEL AVILES

JEAN-PHILIPPE PIERRE

WORCESTER POLYTECHNIC INSTITUTE

A report submitted in partial fulfillment

of the requirements for the degree of

BACHELOR OF SCIENCE in COMPUTER SCIENCE

Advisor:

PROFESSOR EMMANUEL O. AGU

Submitted on: May 5, 2021

Table of Contents

List of Figures

Figure 6.20: Scatter plot showing relationship between sum of questionnaire values and risk score

Abstract

COVID-19 has infected millions over the past year. The virus spreads through close contact with those who are infected. This paper discusses the development of the android app Goatvid Trace which calculates a user's risk of exposure to COVID-19. Our study to test the app found that the mean risk score of WPI students was 25.6%. The paper also discusses the Machine Learning model that estimated distances between two phones with a CV RMSE of 1.587660707.

Acknowledgements

We would like to express our sincere thanks to our advisor Professor Agu for his advice, guidance, and support throughout this project.

We would also like to thank Ermal Toto from Information Technology for providing computing services and helping us with our server issues.

Thank you to Jennifer Hapgood-White from COVID-19 Emergency Response Team for allowing us to interview her to learn about the contact tracing process happening at WPI.

Chapter 1. Introduction

COVID-19 has proven to be one of the biggest challenges the United States has ever faced. This challenge has resulted in more than 30 million infections [13], more than 500 thousand deaths [13], and more than 14% unemployment rate [65]. Both governments and institutions are struggling to create systems and measures that would allow them to reopen and operate under this new normal.

1.1 Background on COVID-19

Coronaviruses are a large diverse group of viruses that get their name from the crown the virus seems to have when looked at on a microscope. The virus that causes COVID-19 is called SARS-CoV-2. It originated in bats and it is the third Coronavirus to be transferred from animal species to humans. SARS stands for Severe Acute Respiratory Syndrome [24]. Signs and symptoms for COVID-19 include: fever, tiredness, chills, muscle pains, cough, loss of taste or smell, difficulty breathing, headache, and sore throat [24].

The disease causes a wide range of signs and symptoms ranging from asymptomatic cases that do not show any signs or symptoms to more severe cases that could result in respiratory infections that could be mortal and various cases in-between. The incubation period of the virus ranges between 2 to 14 days after exposure, with an average of about 5 days. The infectious period starts two days before the onset of signs and symptoms and ends close to 10 days after the onset of signs and symptoms. Asymptomatic patients are infectious for about 10 to 14 days [24].

The peak of signs and symptoms correlate with infectiousness, and the more severe the symptoms, the longer the infectious period is. Even though there are several exceptions, age and pre-existing conditions determine how severe the disease is on them. Old age, diabetes, immune syndromes and respiratory conditions all increase the chance of severe disease [24].

Tests for COVID-19 can either be diagnostic tests or an antibody test. Diagnostic tests are PCR tests that collect a swab from the patient either from the throat or the nose and check for RNA of the virus. The antibody test checks the patient’s blood to look for the immune response to the virus [24].

The virus is transmitted by droplets of saliva that are released while talking, sneezing or coughing, these droplets can enter the body by landing on surfaces people touch and touch their face afterwards or they can be inhaled while they are still in the air. The high rate of infection of the virus combined with the ease of transmission, asymptomatic infectious patients, and the infectious period starting before giving signs result in a highly contagious virus with possibly mortal results [24].

High risk behaviors are those that can increase one's risk of contracting the coronavirus [19]. Some examples include:

- Transportation
    - Using public transportation
- Indoor Gatherings
    - Going to Concerts
    - Attending Religious services
    - Attending Indoor Parties
    - Going to Bars and Nightclubs
- Eating
    - Eating at indoor restaurants
- Exercise
    - Exercising at the gym
    - Playing contact sports
- Work
    - Working in an office
- Services
    - Going to a salon or barbershop

These behaviors are considered risky because they increase one's exposure to saliva droplets which is the method of coronavirus transmission. According to the CDC, behaviors that reduce risk of transmission include wearing masks, limiting travel, limiting time indoors outside your home, and maintaining 6ft of social distance from others [12].

1.2 Tracking High Risk Behavior

Knowing how the risk of contracting COVID is impacted by these behaviors can help people make better and less risky decisions. For example, after learning that public transportation increases one's risk, a person might opt to carpool with a family member. Similarly, after knowing that working in an office can increase your risk, someone might decide to work from home instead.

One of the most powerful, most widely used tools to track these behaviors are smartphones. According to the Pew Research center, 81% of Americans had smartphones in 2019 [53]. Their widespread use in addition to the many built-in sensors, make them useful data-collection tools. This allows smartphones to sense, perform computations, and predict things such as where a user is located, proximity to others, who they are communicating with, and what they are doing. With these data, scientists can study behavior, predict outcomes, and gain a better understanding of people's social networks. Because smartphones are already widely used, smartphone sensing allows for organic data collection that cannot be replicated in a lab [30].

1.3 Introduction to Contact Tracing

Contact tracing is a public health strategy that aims to stop the spread of an infectious disease within a population [24]. It does this by identifying people who may have been exposed and asking them to isolate before they possibly spread it to others. Contact tracing is a vital tool in helping to control the spread of coronavirus in communities [24].

Contact tracing was mostly manual and health workers would manually interview infected people and discover and call their close contacts to inform them. However, due to the advent of coronavirus, public health resources are limited, making it imperative to increase efficiency in contact tracing. As a result, technology has had a larger role in carrying out contact tracing. According to the World Health Organization, using databases to assist contact tracers "[s]treamline[s] the data flow and data management process . . . and improv[es] timeliness of analysis and monitoring" [69]. These databases also serve to keep track of cases, their contact information, as well as any close contacts they may have [69]. This has led to practices such as using central databases to connect contact tracers to cases more quickly and using smartphone apps to report symptoms [74], notify people who have been in contact with a positive case, and record a person's close contacts. Apps such as these make it easier to carry out contact tracing because they are able to record and notify contacts that people might forget to report.

Contact tracing smartphone apps have been increasingly using Bluetooth Low Energy(BLE) to record close contacts. This is because BLE can be used to detect contacts in a more precise and privacy-preserving way than other technologies such as GPS. BLE contact tracing apps are usually implemented by utilizing BLE advertisement packets transmitted by smartphones. Upon receiving a Bluetooth signal, the phones transmit anonymous IDs which are used to notify their users if they have been in contact with a positive case.

Close contacts are so important because COVID-19 can be transmitted through saliva particles that float through the air and thus a close contact could potentially mean exposure to the virus. This means that close contacts are directly related to the risk of contracting the virus and thus it is important to monitor them. Keeping track of them can also allow us to weigh them and calculate a user's potential exposure of the virus.

1.4 Bluetooth Contact Tracing

Contact tracing relies on making note of which people a person was in close contact with in the days prior to becoming symptomatic. Manual contact tracing faces challenges. Often, it is difficult for people to remember all their close contacts. In cases of public transportation or large schools, a person may be in close contact with strangers. This is problematic as those strangers cannot be notified that they may have been exposed. To combat this, Bluetooth has been used to link strangers as close contacts through their phone signals.

1.4.1 How Bluetooth Contact Tracing Works

The main goal of Bluetooth Contact Tracing apps is to make contact tracing more efficient and avoid mass quarantine [41]. The apps achieve this by compiling a list of close contacts and notifying those people if they may have been exposed to the virus.

Phones with a contact tracing app installed can communicate with each other using Bluetooth Low Energy or BLE [8]. The Bluetooth advertisements help other devices find each other. Then, a Central device can begin the connection process. Each phone has an anonymous ID it transmits and exchanges with nearby phones periodically. These IDs are randomly generated and updated frequently to ensure that no one's location or contacts can be traced back to their identity. This process is shown in Figure 1.1.

Figure 1.1: An illustration depicting the generation and use of anonymous ID's [41]

If a person tests positive, they can enter that information into the app. Using their phone's anonymous ID number, the system will reference an anonymous database of contacts and determine who may have been exposed. Those who have been exposed will receive a notification through their app, asking them to isolate immediately. This is shown in Figure 1.2.

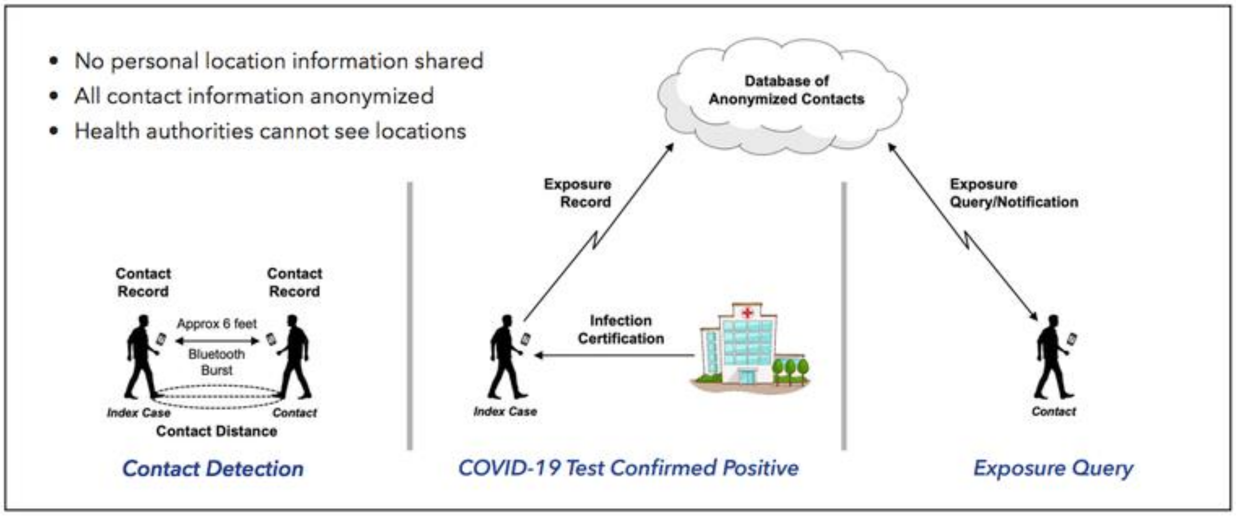


Figure 1.2: A diagram of how Bluetooth contact detection and training works[41]

These apps typically record close contacts and the amount of time spent in close proximity to them. A close contact is typically defined as less than 6ft. BLE can estimate the distance between two phones based on the strength of the signal between them.

1.5 MQP Problem Statement

With millions COVID-19 cases affecting communities worldwide, universities are among the many entities facing the biggest health crisis in their existence. Although regular testing allows organizations to track the spread of the virus, it often is not enough to help individuals prevent their exposure [34]. Students are uninformed about their own risk of exposure to COVID-19 while living on college campuses and a solution is needed to address this issue.

1.6 Challenges

1. Bluetooth Low Energy:

There have been difficulties in implementing approaches to COVID19 contact tracing apps. These apps usually use Bluetooth or GPS to detect contacts. GPS can be used to detect whether someone was in the same building as a positive case, but it fails to detect close contacts because its precision is within a few meters, which is a greater distance than the distance between close contacts. Bluetooth contact tracing apps tend to use the received signal strength (RSSI) from BLE signals to determine distances between contacts. However, numerous factors can affect RSSI values obtained from BLE signals. A study which evaluated the feasibility of using BLE for contact tracing found that BLE signals are often absorbed while a phone is inside a bag or pocket, lowering their RSSI values. The study also observed that the human body can absorb BLE signals and decrease RSSI values. In addition, it found that BLE signals are not absorbed by walling used to separate rooms inside a building, which could potentially lead to false alarms for people who live or work in adjacent rooms [38]. According to another study, different phone models can also influence RSSI values. Different phone models use different Bluetooth hardware, which affects the maximum strength of their signals [39]. BLE contact detection also faces challenges due to data availability. Because COVID-19 is a novel virus, there are not many datasets that contain Bluetooth data for use in contact detection. As a result, there is a limited amount of research in creating predictive models that use BLE signals to accurately detect contacts in real-life environments. During our research, we found no models that were able to use BLE signals to accurately predict distances between smartphone users in non-controlled environments. This lack of prior research makes using BLE to detect contacts a daunting problem to solve.

2. Smartphone constraints:

Using BLE to detect contacts also presents challenges to contact tracing apps due to smartphone hardware. For BLE contact tracing apps to be able to detect contacts at any time, the smartphone has to be constantly transmitting BLE signals. Transmitting Bluetooth signals for prolonged amounts of time can drain the smartphone’s battery which makes contact tracing apps more difficult for users to adopt. Smartphone operating systems may also limit the amount of time that phones can transmit or receive Bluetooth signals for, making it more difficult for developers to create contact tracing apps.

3. Privacy concerns:

Privacy concerns also make contact tracing apps solutions a difficult task. User information such as demographic information, user habits, and location make it easier for apps to carry out contact tracing at the expense of user privacy. As a result, a significant number of contact tracing apps focus on preserving user privacy by using increased security measures during data transmission and limiting the amount of user information collected. However, using less information limits can limit the effectiveness of a contact tracing app. For example, a contact tracing app with access to users' location could potentially inform users about crowded areas and assign more risk to users that visit crowded locations. Apps without access to location data would not be able to have such a feature. Therefore, the tradeoff between user privacy and functionality presents a challenge to contact tracing apps.

4. Novelty:

Another challenge is that approaches to estimating the risk of contracting COVID-19 are about as novel as the virus itself. Relatively few COVID-19 contact tracing apps estimate their users' risk of being infected. In our research of thirty contact tracing apps, only two of them attempted to estimate the user's risk of being infected with coronavirus. As a result, estimating the risk of being infected with COVID19 presents a challenge to this project and to contact tracing apps as a whole because its status as a novel and less known problem makes it more difficult to solve. There is not much existing information about how COVID-19 risk can be estimated, making it harder to create such a feature in contact tracing apps.

1.7 Overview of Goatvid Trace

To address this problem statement, we created Goatvid Trace, an Android mobile application aimed towards the WPI student population. Goatvid Trace calculates a risk score for a user which indicated their likelihood of being exposed to COVID-19. For example, a higher risk score means that the user engaged in risky behaviors and now has a higher risk of being exposed. To calculate this, the app records 'close contacts' or interactions between two users where they are closer than 6 ft for 15 minutes or longer. If these interactions occur at a certain GPS tagged location such as a restaurant or bar, the risk associated with that interaction is considered to be higher. The risk score also accounts for behavioral data that is inputted by the user through an in-app questionnaire. These behaviors are: going to class, getting food, visiting campus, mask wearing, social distancing, and transportation. In addition, the app also displayed COVID related statistics for their area and provides a messaging service for Health Services to communicate directly with students about COVID updates and reminders. Figure 1.3 below shows an overview of our proposed mobile app.

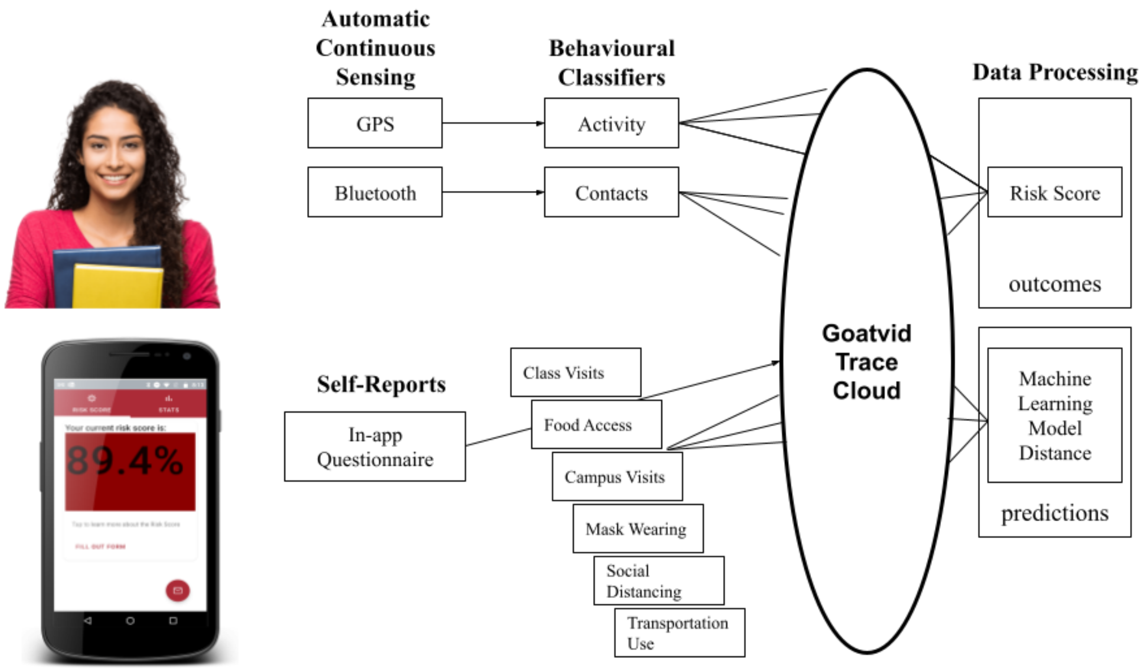


Figure 1.3: Overview of the Goatvid Trace mobile app

1.8 Previous COVID Mobile App Approaches

The worldwide impact of COVID-19 has led to numerous attempts to solve this problem. Numerous approaches to COVID-19 consist of a smartphone app that carries out contact tracing. Usually, the scope of the contact tracing varies by app. Apps such as Hansel conduct contact tracing for the general population, while other apps conduct contact tracing for specific communities. For example, the app CoronaWarn carries out contact tracing specifically for people in Germany. Regardless of the scope of their contact tracing, all of these apps detect close contacts, usually through GPS or BLE. They also allow users to report themselves as a positive case and inform users if any of their close contacts were found to be a positive case.

Some of these apps offer additional features. One such feature is social distancing enforcement. Mind the Gap is an app that uses Bluetooth and high frequency audio signals to estimate distances between phones. It also reminds users to maintain social distancing when they are within a certain distance of each other. Other apps such as Private Tracer can be used as a channel to disseminate public health information so that users are informed about COVID-19.

Among these features, COVID-19 risk estimation is relatively uncommon. Only two of the thirty apps we researched offered COVID-19 risk estimation. One app utilized data from the user's contacts to estimate risk, while the other estimated risk by modeling exhaled clouds of the virus. The small number of apps that offered this feature motivated us to implement this feature in our own project.

1.9 Goal of this MQP

The overarching goal of this MQP is to determine the effectiveness of using a smartphone passively monitoring a user's behavior to calculate risk of coronavirus infection based on sensed and inputted behaviors. This goal was motivated by the lack of apps found that offered this feature. COVID-19's high rate of infection leads to possibly dire consequences for high risk behaviors. As a result, it is important for users to know about their personal risk of infection so that they can make informed decisions. Because most apps we researched did not calculate risk of infection, we desired for our app to do so. We also decided on this goal so that students at WPI are more informed about COVID-19 and about behaviors that increase their risk of contracting the virus so that they can choose to partake in less risky activities. Therefore, we aim to implement an intelligent smartphone application that:

- Calculates an estimated personal risk score that reflects potential exposure
- Uses Machine Learning to accurately determine distance between users
- Facilitates anonymous contact detection using BLE
- Serves as a tool for WPI Health Services to communicate with the student population about COVID related information and updates
- Displays relevant community COVID statistics

1.10 MQP Roadmap

The rest of the MQP report is as follows: Chapter 2 describes the background information required to understand our work. Chapter 3 explores related works including other apps and studies to provide context. In Chapter 4, we detail our proposed app design broken down into modules.

Chapter 5 describes how we implemented the app design which includes specific technologies and tools. Chapter 6 presents our results from our Machine Learning model, the Beacon distance library as well as the user study we conducted to test the app. These results and future work are discussed in Chapter 7. Finally, Chapter 8 presents our conclusions for the project.

Chapter 2. Background

2.1 Contact Tracing

Contact tracing is a public health strategy that aims to reduce the spread of an infectious disease. By intervening, a contact tracer can stop a person with the infectious disease from passing it on to others. In the long run, this has a big impact on the spread of the disease [24].

2.1.1 Steps to Investigate Cases and their Contacts

When a person's coronavirus test comes back positive, they will be contacted by a contact tracer in their community. A person who tests positive is called a case. The case will be asked to go into isolation. Isolation is when a sick person avoids contact with others for the duration of their illness to avoid passing it on. A person with coronavirus will have to isolate until they have met the following conditions: 10 days have passed since they had symptoms, all of their symptoms are improving, and they have been without fever for 3 days without medication.

Next, the contact tracer aims to identify all of the people the case has had contact with during their infectious period. Because a person with coronavirus can be infectious and spread the virus for up to two days before becoming symptomatic, it is imperative for those who have been exposed to the case to quarantine before they spread the virus to additional people.

To identify those who a case has been in contact with, the contact tracer will conduct an informal phone interview. They will ask the case what they did the past couple days, and who they saw. Sometimes, they suggest that the case looks through their phone and social media to make sure they have remembered all interactions. Using the public health guidelines in their area, the contact tracer determines which of those people are considered contacts and have a higher risk of being exposed to the coronavirus.

Similar to isolation, a contact is asked to quarantine themselves. Although they may not have symptoms of illness yet, they are asked to stay home and avoid contact with others to avoid transmitting the virus if they have it. A contact will be asked to quarantine for 14 days as long as they do not develop symptoms. If they do develop symptoms, they will be considered to be isolating.

Throughout this process, the contact tracers will check up with both cases and contacts. They monitor the case's symptoms and help arrange for medical attention if they begin to experience more severe medical attention. The contact tracers also help to remind both cases and contacts why it is important for them to isolate or quarantine. [24].

2.1.2 The Ethics of Contact Tracing

Although contact tracing is a common public health tool, there are a lot of legal issues that arise from it regarding the privacy of the cases and the confidentiality of their information.

A contact tracer can ask for private information only for the purposes of protecting the public. Similarly, a contact tracing app can only learn about medical information specific to the contact tracing work it is performing.

A public health intervention is legal if it respects the individuals and their rights. It is essential that everyone is treated equally and fairly, regardless of who they are. Additionally, the intervention must be a benefit to society. [24].

Our application requests only the information that is required to calculate a risk score. This ensures that all the information is being collected ethically.

2.1.3 Use of Technological Tools

Using technology to run an intervention is useful since public health resources are limited. By using technology, public health officials are able to reach people quickly while improving efficiency.

The tools used for contract tracing include a central database that stores a list of people and their information. This database can report cases to investigators automatically while reducing the time between diagnosis and call. Another tool used is several symptom tracking apps that allow users to manually enter information. Also, apps may use text messages in order to remind contacts to quarantine in case they have been close to a person that reported that they are ill.

When identifying contacts, there might be some problems that apps can solve. For example, a person might not remember all their contacts or may not want to talk about their contacts because of privacy concerns. Also, there is a high possibility that cases may not know the phone numbers for contacts or the phone numbers may be incorrect.

In countries such as China and South Korea, the government can access the smartphones of citizens and store their information in a centralized database. The advantage of having this kind of database is that all the contact identification and contact tracing are done immediately. However, this raises some privacy concerns as people need to share their location with the government in order for this solution to work. Finally, GPS technology is not a good approach because a person could have been in the same shop with someone, but might not have been exposed to everyone in the shop[42].

In the US, several apps that use Bluetooth technology (BLE) have been developed in order to do contact tracing. Those apps do not collect private or confidential information. They usually keep track of which other phones have been in contact with a person and in case that person falls sick they notify all their close contacts to quarantine [24].

2.1.4 Contact Tracing and Goatvid Trace

Although the Goatvid Trace application is inspired by contact tracing for COVID-19, the app itself does not perform contact tracing. Instead, the app detects contacts. This means that information about a user's contacts is never shared with anyone. It is only used to estimate their risk.

2.2 Bluetooth Contact Tracing Apps

In this section, we will discuss common Bluetooth contact tracing systems, their efficacy, and challenges currently faced.

2.2.2 Apple and Google API (ENS)

The Exposure Notification System of ENS is an API created in a joint effort between Apple and Google to help slow the spread of the coronavirus. This tool is meant to be used when developing contact tracing apps for specific communities, it is not a contact tracing app itself. It uses Bluetooth to determine a user's contacts and record them locally. All users are represented by an anonymous ID that changes often so a user cannot be tracked to traced by Apple or Google. It also allows users to opt-out at any time. Although this seems like an ideal solution, ENS is only available to public health authorities and the local governments associated with them. As an MQP group creating a proof-of-concept app, ENS is not the right fit for us as a contact tracing API [27].

2.2.3. Do Bluetooth Contract Tracing Apps work?

There is much focus on the development of contract tracing mobile apps. However, it is important to first determine whether these apps are able to effectively reduce the spread of coronavirus. The biggest limitation of contact tracing apps is that an app is only able to record the interaction with another person who has downloaded the app. For this reason, the more people who download the app, the more interactions that will be recorded, and the more effective the app will be. However, Farzad Mostasharia, the former national coordinator for health information technology at the U.S. Department of Health and Human Services, notes that “even if 1/3 of the population downloads and uses a contact-tracing app, it will still only cover about nine percent of close interactions” [41]. A graph explaining the likelihood that a given interaction or contact between two people will be captured by these contact tracing apps can be seen below in Figure 2.1.

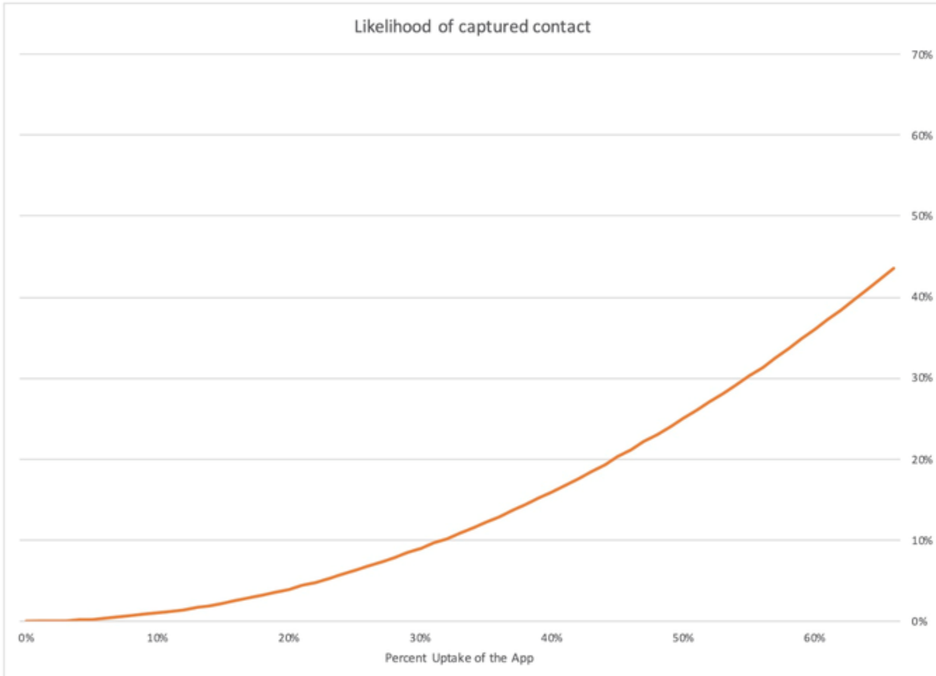


Figure 2.1: A graph showing how the likelihood of captured contacts changes with the percentage of the population that uses a contact tracing app [41]

For example, TraceTogether is a contact tracing app for Singapore [42]. Although it has been downloaded by over 1.1 million people, this only accounts for around 20% of the population. As a result, there is only a 4% chance that given an interaction between two people, both will have the app on their phones. Additionally, these apps do not account for the portion of the population who do not have smartphones. These populations are typically older people or migrant workers, two groups who are already at increased risk. In Singapore, migrant workers make up a large portion of coronavirus cases due to crowded living conditions.

It is important to understand the community that a contact tracing app is being developed for and think about whether it will meet the community's needs.

2.2.4 Bluetooth Contact Tracing Apps and Goatvid Trace

Goatvid Trace, which also used Bluetooth, faced many of the same problems as other Bluetooth Contact Tracing apps. Like previously stated, our app was not eligible for ENS so

Bluetooth contact tracing was implemented manually. In addition, Goatvid Trace aimed to serve the population of WPI students. However, unless a majority of students have the app installed, most close contact interactions would not be recorded.

2.3 Machine Learning

We utilized machine learning to analyze smartphone Bluetooth signals in order to determine subject proximity to determine user contacts. Contact tracing applications can use machine learning to facilitate their features. For example, COVI, a contact tracing application that was proposed in Canada, used machine learning to estimate risk of infection between contacts [4]. It did so by creating a model that used users' reported symptoms, demographic information, and contacts to predict their risk of COVID-19 contagiousness.

According to Alpaydin's *Introduction to Machine Learning*, machine learning is the process of "programming computers to optimize a performance criterion using example data or past experience" [3]. The result of this process is the creation of a statistical model from a dataset. The model can then be used to generalize the solution to a problem. Machine learning creates this model by starting with a model structure and finding the best fit of this model to the example data it was provided with. It uses a set of inputs called features to determine the qualities of the data it will use to find this fit, and the values it will use to solve the problem after the model has been created. The process of fitting the model to the example data is called training. While a model is being trained, it will adjust itself based on the values in each record in the example data. There are two types of machine learning: supervised learning and unsupervised learning. Supervised machine learning uses example data such that each data record is labelled with the solution to the problem (also known as the target label) the model is intended to solve. Conversely, example data for unsupervised learning does not contain these labels. Since this project solely utilizes supervised learning, this section will only discuss supervised learning [3].

Supervised learning models consist of a hypothesis and a cost function. The hypothesis is a mathematical equation used to calculate the model's output for a certain input. It consists of the model's features and a parameter corresponding to each feature. The model uses the parameters to weight each feature. The model adjusts itself during training by changing the values of these parameters. The cost function is a function that calculates the amount of error between the model's predictions and the correct values to be predicted. As a result, algorithms that train supervised machine learning models attempt to minimize the value of the cost function [3].

There are numerous types of supervised learning models. Different types of models vary in the ways that they make predictions and the cost functions that they use. Two common types of supervised learning are regression and classification. Regression models predict a continuous value that is calculated by their hypothesis. In the case of linear regression, the hypothesis consists of a weighted sum of the model's features and their parameters. However, there are numerous types of regression algorithms that use different hypotheses.

Machine learning classification: models predict a value from a set of discrete values, and are used to make classifications about their input. An example of a classification model is a logistic regression model. In the case of binary classification, a logistic regression model will use a hypothesis similar to one used by a linear regression model. However, it will output a binary value by outputting 1 if the value of the hypothesis is above a given threshold, and 0 otherwise [3]. We briefly considered using logistic regression in our project, but decided to use linear regression instead.

Machine learning regression: Apart from linear regression, many other machine learning algorithms predict a continuous value. We will outline those that apply to our project. One such machine learning algorithm is a decision tree model. Like linear regression models, decision tree models can be used to predict a continuous value. However, decision tree models do not use the weighted sum of the model's features as a hypothesis. Instead, they use a tree structure to make predictions. Each node of the tree has branches based on conditions concerning the input. The leaves of the tree represent predictions. A common regression algorithm that uses decision trees is the random forests regression algorithm. Random forests models consist of numerous decision tree models whose predictions are combined to result in a single prediction. The decision trees in a random forests model use random samples of the training data, and are formed using random sets of features [15]. Another example of a linear regression algorithm that uses multiple decision trees to make a prediction is XGBoost [16].

Support vector machine (SVM) regression models use an algorithm to attempt to define a plane that separates sets of data points within the dataset so that the distance between the plane and any data point is maximised [7]. Machine learning models can also use a K nearest neighbors algorithm for regression. The K nearest neighbors algorithm trains a machine learning model so that it can make a prediction about a certain input from a certain number of training data points that are most similar to the input [63].

Figure 2.2 illustrates the types of machine learning and the algorithms previously outlined.

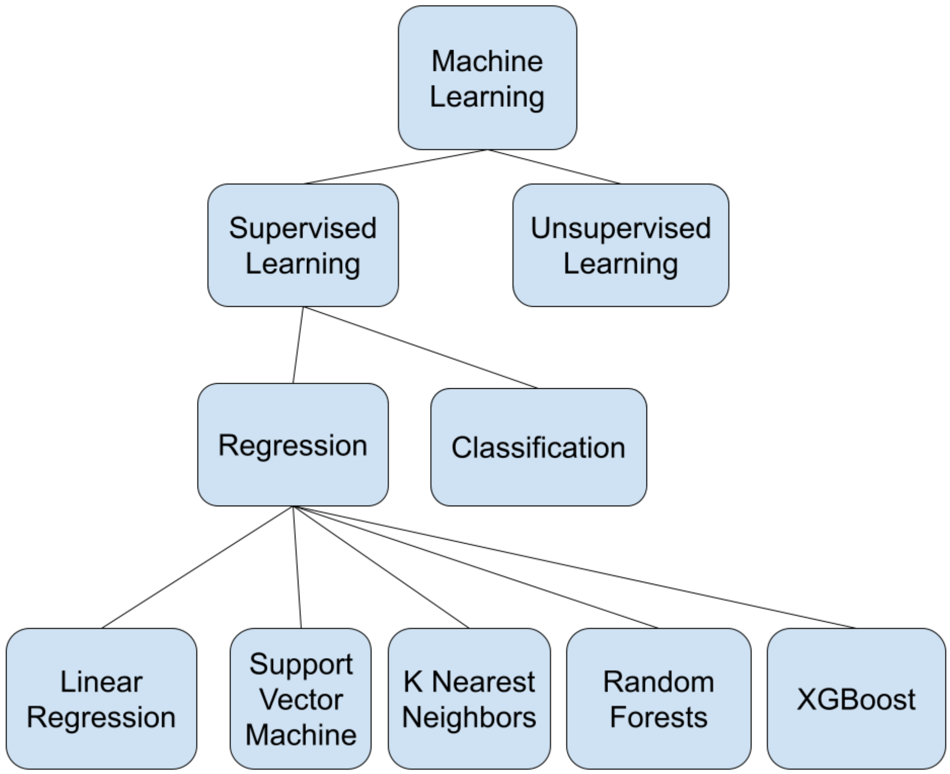


Figure 2.2: Diagram illustrating of types of machine learning [3]

2.3.1 Machine Learning and Goatvid Trace

Goatvid-Trace also made use of machine learning. The app used it to create a model that used BLE signals as input to estimate distances between smartphones. As a result, our research about supervised learning and regression models was critical to the development of this model.

2.4 Privacy preservation and data protection

Smartphones frequently request access to our personal data. Even though this makes our phones powerful tools through which we can do almost anything from financial transactions to private communication, this also means that a breach in security could have catastrophic consequences for its user. The average cost per cyber-attack is calculated to be about 500 thousand

USD [32]. This is why the growth of the mobile application market and its usage has gone hand in hand with ensuring security and data privacy.

However, the fact that this market is growing has proven to be another obstacle, especially for Android devices. The massive number of apps available and being produced has made it impractical for app markets to verify every app and qualify them as non-malicious. Adding to this that there are also third-party app markets, users can never be completely sure what apps could represent a security breach. Thus, the way each app handles and protects your data has never been more important [57].

2.4.1 Privacy preservation and data protection and Goatvid Trace

There are numerous privacy preserving protocols and resource managers that allow developers to produce privacy preserving applications without compromising much of the application's functionality. However, since we are producing the app from scratch, we have decided to implement design strategies to protect the users' data.

More specifically we have selected a hide privacy design strategy, which is defined to "protect personal data, or make them un-linkable or unobservable", "[p]revent personal data becoming public", and "[p]revent exposure of personal data by restricting access, or hiding its very existence" [25]. To achieve the correct implementation of the strategy and its efficiency we will:

- Prevent unauthorized access to personal data (Restrict):
  - User data is stored on the database and is not available to other users.
- Remove the correlation between pieces of personal data (Dissociate)
  - Each user was assigned a subjectID during onboarding and the only excel sheet containing the match between user and subjectID was deleted
- Encrypt the data that is posted to the rest API
  - Each request is protected using cryptographic protocols designed to provide communications security

Chapter 3. Related Work

3.1 Existing COVID-19 Contact Tracing Mobile Applications

While reviewing the existing apps that address the COVID-19 pandemic, we found that many of the apps had similar purposes. We grouped these apps into four categories: contact tracing within the general population, contact tracing for a specific community, social distance enforcement, and public health information dissemination.

3.1.1 Contact Tracing within the General Population

Regarding the contact tracing within the regular population, we found 4 applications that aim to build a contact tracing system for the general public. Those are: Covid-id, Hansel, C19X, and CovidWatch.

Covi-id is a privacy-preserving cross platform application (Android, iOS support) which does risk management by using QR code scanning. It notifies users if they have come into contact with someone with Covid-19. Even if users do not have phones, it still provides the option to register and print your unique QR code using a friend's or relative's phone [18].

Hansel is another GPS-based solution with a simple interface that gives users the ability to report a case, start or stop tracing with the promise of encrypting your location or user data. It uploads a hash of your location and time which matches users to other people that have been in the same place or have crossed paths with them [1].

C19X is a cross platform app that uses BLE and SHA technologies which enable autonomous and secure contact tracing on many devices. Based on their GitHub information, they provide an Android and an iOS application which they maintain independently. The app collects Bluetooth Beacon Data in order to accomplish its goal [9].

Covid Watch is another application that was implemented at the University of Arizona. It claims that if someone comes into contact with a user and chooses to enter their positive test results it will alert that user anonymously. Also, the app claims that it is calculating the estimated personal risk level of a user, which is a concept that is going to be discussed later [67].

We implemented contact tracing using machine learning in our app. We decided to incorporate some of the features of these apps into our project. Some of these apps use BLE, which we decided to use for contact detection. These apps also tend to try to protect users' data through increased security and privacy measures. Because of this, we sought for our app to do this as well.

3.1.2 Contact Tracing for Specific Communities

Because the coronavirus is spread through close contacts, it spreads throughout a community easily. To fight this, many apps have been developed to assist with contact tracing in specific communities. Some of these communities include: towns, states, countries, schools, or even large office parks.

For example, in Germany a contract tracing app called CoronaWarn was commissioned. This app, and many like it use Bluetooth to identify contacts of a given user. This is done by assigning each user a unique and anonymous identifier. The app collects the unique identifiers of contacts based on distance calculated and time spent in close proximity. Often, users can input a verified positive test result into their app and the app will anonymously notify the contacts of the positive user. In other cases, the person who tested positive could upload their contact history to a database for contact tracing professionals to act upon [23].

Many of these apps are open source to create trust and an assurance of privacy for the community it serves. Some other examples of these kinds of apps are Hamagen for Israel, Pan-European Privacy-Preserving Proximity Tracing (PEPP-PT) for Europe and CovidWatch which was implemented at the University of Arizona [33] [51]. These apps are displayed in Figure 3.1.

Name	Maker	Goal	Functionality
CoronaWarn [23]	SAP and Deutsche Telekom	Slow the spread in Germany	An app that enables you to retrieve test results electronically, and it helps to identify possible exposures you have had to people diagnosed with COVID-19
Hamagen [33]	Israel's Ministry of Health	Slow the spread in Israel	Privacy ensured contact tracing app that informs people of possible exposure for Israel
PEPP-PT [51]	Not Disclosed	Not Disclosed	Not disclosed
CovidWatch [67]	Covid Watch	Have an anonymous exposure notification app	Free and Anonymous Exposure Notification App (implemented at University of Arizona)

Figure 3.1: Contact tracing apps aimed at specific communities

3.1.3 Social Distance Enforcement

In addition to contact tracing, social distancing is an important measure to help stop the spread of coronavirus in communities. As a result, there are mobile applications that aim to help enforce social distancing guidelines by notifying users when they go too close to each other.

One example of this category of apps is: Mind the Gap. This app was developed by software developers and sensor experts from HackPartners and Network Rail. Their goal was to provide a way for essential workers to work during the pandemic, while still maintaining their safety and privacy. The Mind the Gap app uses Bluetooth and high frequency audio signals to estimate the distance between two phones with 10cm precision. When two users with the app come within a certain predetermined distance of each other, the user is alerted to remind them to remain socially distant. The app works between both iOS and Android phones and it is aimed at offices and other work environments where all employees can download the app. 1point5 is another app that notifies users if people have breached their circle of 6 feet or 1.5 meters [28].

Other apps such as Crowdless focus on informing users of how crowded a location is before they go there. This helps users to make decisions on where to go grocery shopping for example based on their ability to socially distance at a given location. These apps calculate the crowdedness of a location from user input, existing data, and machine learning. This app is not specifically aimed at stopping the spread of COVID-19, but it certainly can be used that way [20].

We briefly considered implementing a social distance enforcement in our app, but decided it was outside the scope of our project. The social distancing apps covered in this section are shown in Figure 3.2.

Name	Maker	Goal	Functionality
Mind the Gap[28]	Hack Partners	Enforce Social Distancing in an office	Monthly subscription based mobile app that alerts users in office settings when they are not social distancing with a sound.
Crowdless [20]	Lanterne	Inform users to take decisions when trying to avoid crowded places	Uses user data to measure how crowded places are and informs the users.

Figure 3.2: Social distancing apps

3.1.4 Public Health Information Dissemination

There are 4 applications that belong to the category of applications that are disseminating information. Those are: Private Tracer, Contact Tracing Covid19, Global Epidemic Prevention Platform and CovidSafe.

Private Tracer is an app that works in the Netherlands and cooperates with the Ministry of Health [45]. Their goal is to find out if an application can be a useful tool in fighting the covid-19 pandemic. The features of the application include proximity tracking and notifications. The Global Epidemic Prevention Platform application is working in a similar way (informs the Ghanian government and notifies users at risk). The rest of the applications are using similar technologies (BLE, GPS, ENS).

After seeing these apps during our research, we decided to incorporate a health information dissemination feature into our app. We believed this feature was important to WPI's needs because it would enable WPI Health Services to send out coronavirus-related information more quickly.

3.1.5 Risk Score Calculation

Our COVID-19 contact tracing app calculated a smartphone user's risk of infection. As a result, it was beneficial to our project to look at other apps that have this feature, and how they calculated risk. There were 2 apps that attempted to calculate the user's risk of being infected with COVID19. These apps were Corona-Warn-App and Covidwatch.

Corona-Warn-App is an app used to facilitate contact tracing in Germany [17]. It attempts to let users know whether they have been exposed in a decentralized manner. It uses BLE to determine proximity between users and then notifies users if they have been exposed to a positive case. What makes this app unique when compared to the others is that it uses a risk score to determine whether the user should be notified. Each contact has a risk score that is calculated using the number of days since the user has been exposed to a positive case, the duration of the contact, and the distance from the contact. These scores are aggregated to create a total risk score. Then, for each encounter, their durations are aggregated, weighted by their distances, to create an exposure score. After the total risk score is normalized, it is combined with the exposure score to create the final risk score.

CovidWatch is a contact tracing app being used in the University of Arizona [67]. It carries out contact tracing by using BLE to detect contacts, and it notifies users if they have been in contact with a positive case. It instructs the user on what to do based on the risk score of the case the user was exposed to. The risk score is calculated using a model of the shape and orientation of an exhaled cloud of the virus, and the expected amount of disease the user would inhale, depending on the infectiousness of the contact [68]. This risk score decreases based on how long the user has been free of symptoms after exposure.

CovidWatch uses a risk score to inform how risky an interaction with an infected person was, while Goatvid Trace uses a risk score to generate a general risk of exposure based on all interactions. Both CovidWatch and Goatvid Trace use Bluetooth signals to estimate distances between two users. Because CovidWatch also serves as a contact tracing app, it uses information about an infected person's contagiousness at the time of the interaction to further inform the risk score. Alternatively, Goatvid Trace incorporates behavioral information like transportation and mask wearing habits to further inform its risk score. It also uses past risk scores to create a rolling average.

3.2 Use of Machine Learning in BLE Distance Calculation

We used machine learning to calculate the distances for contact tracing. As a result, we researched how machine learning has been used to calculate distances from BLE signals, so that we could determine how to apply it in our project. This section will outline two studies related to distance estimation from BLE signals. It should be noted that both of these studies used data in controlled environments. In these environments, there were no physical objects obstructing the recorded BLE signals and no other BLE signals that could have interfered with the ones being recorded.

One study attempted to find a machine learning model that best fitted a dataset containing distances and corresponding RSSI values of BLE signals received at those distances [60]. Its experimental setup consisted of a smartphone and a Bluetooth beacon. The study collected data at distances ranging from 0.5 to 3 meters and at angles ranging from 0 to 180 degrees. After obtaining this dataset, the study used it to train linear regression models with a polynomial, power, and exponential structure. Each of the models used a linear combination of an RSSI value as input. The study concluded that the polynomial model was the best fit for the dataset, with an error rate of 25.7%.

A device can send BLE signals with different intensities. The intensity of a BLE signal is called the transmission power. BLE signals at higher transmission powers can be detected within larger ranges, but use more power to transmit. Another study attempted to observe whether using the transmission power improved models for BLE distance estimation. The study collected RSSI data for BLE signals between phones at distances ranging from 0.5 to 22 meters and at different transmission power levels. Part of this study consisted of comparing two linear regression models trained on this data: one which used the transmission power and RSSI as inputs and another which only used RSSI. It found that using transmission power in the model lowered the error rates for linear regression at all transmission power levels except high [26]. Figure 3.3 displays this actual distance against predicted distance for this model.

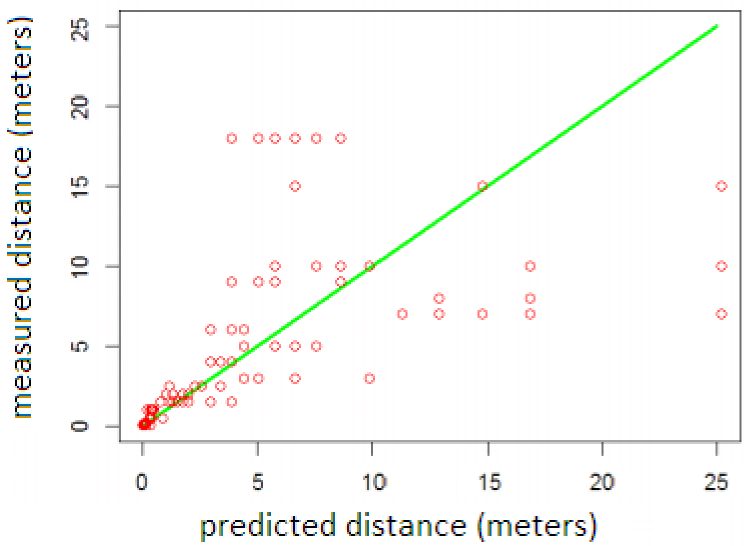


Figure 3.3: The actual distance versus predicted distance for the model using transmission power [26].

We did not incorporate this research into our project because it would not suit contact detection purposes. These experiments took place in controlled environments free of obstacles, while contact detection takes place in environments that are potentially filled with obstacles. However, these projects motivated us to research whether more complex model structures could be used to carry out BLE distance estimation in everyday environments.

3.3 Cough Detection

Cough detection is a field where audio and other sensors are used to detect coughs, often to screen for disease. Because cough is a symptom of COVID, this cough detection method could be used to sense symptomatic COVID people nearby [14]. We have researched other cough detection methods and software to see if cough detection is a viable option for detecting exposure or illness from COVID.

3.3.1 Examples of Cough Detection Software

Cough detection software has been used to detect the number and frequency of coughs in people diagnosed with chronic caught. In a 2018 study from Beihang University, patients were fitted with microphones to detect the number of coughs they had per day in order to understand the severity of their chronic cough. However, this study was more focused on the number of coughs over a period of time, not a single instance of a cough indicating disease like would be in the case for COVID [62].

Another study from the University of London used smart watches to detect coughs from a user. The smart watch would turn on its microphone and record when it sensed the accelerometer of the watch moving in a way that indicated a cough. The accelerometer would be triggered when the user rapidly pulled their hand up to their face to cover their mouth before a cough. This study seems like a great solution to the privacy concerns that audio recording often brings. Unfortunately, the intended population of our app, WPI students, do not all have smartwatches, so it is not an effective solution for our mobile application [50].

3.3.2 MFCC and Scene Classification

MFCC or Mel Frequency Cepstral Coefficients have “features [that] represent phonemes (distinct units of sound) as the shape of the vocal tract (which is responsible for sound generation) is manifest in them” [49]. This allows for sounds to be identified by the ‘shape’ they are. MFCC features are often classified using machine learning and neural networks such as Convolutional Neural Networks (CNNs) a machine learning network typically used for image analysis.

One example of MFCC being used successfully in cough detection was in a study to diagnose pertussis. Due to the unique nature of pertussis (also known as whooping cough), the researchers were able to diagnose all cases of pertussis in the study without any false positives. This study used a relatively low-cost algorithm and suggested its use in smartphone applications.

3.3.3 Cough Detection as Future Work

Although there is a lot of interesting work that can be done with MFCC's and cough detection with COVID, we felt that this aspect would be a big undertaking and is significantly outside of the scope of our proposed application as a contact tracing and risk assessment application. Additionally, a psychological study at the University of Michigan found that people are inaccurate detectors of whether a cough is caused by illness or environmental factors [44]. For future work, we would be interested to see if coughs specific to COVID can be detected and if so, is it an accurate indicator of exposure or current illness.

3.4 Sensing User Health Through Smartphone Sensors

Smartphone sensors and the data they can record present an opportunity to make observations about users' behavior. Such sensors include accelerometers, Bluetooth receivers, GPS, light sensors, and microphones. User behavior data such as level of physical activity, social interaction, and other activities taken by the user can be derived from the signals given by smartphone sensors [29].

One of the ways that user behavior data has been used in research is to detect changes in the smartphone user's health. For example, an MIT study was able to find a relationship between illness symptoms and user behavior. At the beginning of the study, a group of students were given smartphones that continuously recorded Bluetooth interactions between phones, WLAN location data, call records and text message records. The participants also reported their symptoms on a daily basis. This was used to create a database of symptoms and smartphone sensor data. The study found that participants with a runny nose showed a higher amount of calls and text messages in total and late at night, and that participants with a sore throat displayed more Bluetooth interactions with other members of the dormitory. Participants with influenza also had lower amounts of late-night phone communications, late-night Bluetooth interactions, and WLAN locations recorded. The study was able to use the user behavior and self-reported symptom data to create a Bayesian-network classifier that used user behavior as input to detect symptoms. It was able to do this with an accuracy ranging from 60%-80% [6].

Another study used a similar method to find a correlation between mental wellbeing and the amount of physical activity and sleep of the user. Participants were given smartphones and asked to self-report their energy and mood daily. Throughout the study, the smartphones would record accelerometer data to measure the user's level of physical activity and the amount of time the user slept for each day. The study found that participants with greater amounts of sleep and daily activity self-reported better moods than other participants. It was also able to use this data to create a predictive model that predicted a user's mood and energy levels based on their physical activity and sleep [22].

Although this project does not attempt to detect symptoms in users, this research is relevant to our project because it gives us insight into how smartphone sensors can be used to assess user behavior. Understanding user behavior helps us to estimate users' risk.

Chapter 4. Proposed Goatvid Trace App Design

In this section, we will outline our proposed mobile application, Goatvid Trace. Goatvid Trace aims to provide users with information about their risk of contracting COVID-19. Currently, Goatvid Trace is aimed towards students at Worcester Polytechnic Institute (WPI). Users are able to see both their personal risk score as well as stats about COVID-19 transmission within the WPI community.

4.1 Risk Score Calculation

The app utilizes a formula to calculate each user's potential exposure to COVID-19, the formula is meant to employ a holistic approach by analyzing each user's behavior and habits and using them to estimate how much these increase its chances of contracting the virus. By breaking down how often and why the user leaves their home we are able to provide what we call a risk score. This risk score is a measure in percentage of how likely a user is to be in presence of the virus and possibly get infected. The user can use this score as an input to change or tweak its routine in order to get a lower score and thus lower its chances of catching the disease. This is shown in Figure 4.1.

Inputs and Outputs for the Risk Calculation Formula	
Inputs used for Risk Calculation	
Input	Rationale
Questionnaire response values	The questionnaire contains 6 questions each of which provides an insight into how the user's routine looks like. Aspects including but not limited to: how they get their food, how they move around(transportation), how they attend classes and how often they go out and why. The idea here is that we are tracking close contacts and these behaviors are likely to cause a user to get close contacts. Thus, we want to give a numeric value to the user's routine so the daily close contacts can be complemented and the calculation more accurate. The lowest score a user can get in the questionnaire is 4 and the maximum is 37 by choosing the highest or lowest response values respectively.

Close Contacts the user had in the last 24 hours	Close contacts are all given a value of one and then weighted by the tag from the place's API tag values in appendix C. This is meant to give more weight to contacts occurring on places that have been recommended to be avoided during the pandemic. Under the assumption that in those places there's a higher chance of being exposed to the virus.
Last calculated Risk Score	The formula uses the last risk score calculated to average the value of the new risk score when calculating it. The idea behind this is to avoid having a risk score that's constantly flipping between high and low from one day to another.
Output produced by Risk Calculation	
Output	Rationale
Risk Score	The Calculation produces a risk score between 0 and 100 inclusive. Because we cannot ensure complete certainty 0 does not mean not infected and 100 does not mean infected. Rather the score is a percentage of how likely the user is to have been exposed to the virus. It's worth to clear that exposure does not necessarily mean infection.

Figure 4.1: A table showing the inputs and outputs of the risk score

The three inputs of the Risk Score calculation can be further broken down into their individual components:

First, the questionnaire is a complementary method for the passively measured close contacts. These questions account for user's habits that contribute to a higher risk of exposure, as shown in Figure 4.2.

Questionnaire questions:	
Question:	Rationale:
Class Value	This question provides an insight into how the user attends its classes. In person classes often mean sitting in a closed space with several people thus providing an opportunity for potential exposure.
Eating Value	Getting food provides various potential exposure situations, including:touching surfaces to grab items at the Grocery store and sitting in an indoor restaurant maskless while dining. This question allows us to assess how often the user is in these situations.

Campus Visits Value	The wide range of extracurricular activities at WPI makes it possible for a user to take all their classes online and still visit campus frequently. This question takes such scenarios into account.
Mask Wearing Value	This question is a self-assessment of how well each user is following the CDC recommendations regarding wearing face coverings while outside or in the presence of other people.
Social Distancing Value	This question is a self-assessment of how well each user is following the CDC recommendations regarding social distancing while in the presence of other people.
Transportation Value	Using transportation could mean sitting in a crowded train for several hours or driving with a friend. This question takes those risks of exposure into consideration.

Figure 4.2: A table showing the questionnaire inputs of the risk score

Next, is the close contacts portion of the risk score. They provide specific and continuous information about a user's risk of exposure. Because the app is constantly searching close contacts with real users, the data from this input is as accurate as possible. Figure 4.3 describes how close contacts are weighted.

Close Contacts and their Weighing:
Rationale:
Close contacts are used in contact tracing strategies because of the high likeliness of exposure from each of them. However, other factors also play a part on how likely exposure is such as if it's a closed environment or a ventilated one or how crowded the surroundings are. This is why we will weigh each close contact by the place type where it was recorded. Accounting thus for both close contacts and types of places visited.

Figure 4.3: A diagram showing how close contacts are weighted

The final input, the Rolling average is calculated by using the last calculated risk score and averaging it with the new one to connect all the risk scores and avoid drastic changes in daily risk scores that would make each calculation more inaccurate.

In order to calculate one's personal risk score, the app uses the three inputs as values in a formula to produce the output which is the Personal Risk Score. This is a percentage where the higher it is, the higher the user's predicted risk of being exposed to COVID. The algorithm's design is displayed in the pseudocode below:

```
Function Get Questionnaire Value(SubjectID):
    qResponse = GET questionnaire entry from Database with
     SubjectID;
    /* This get request takes the SubjectID as input and
       returns the questionnaire response values stored with
       that SubjectID from the database in a response      */
    questionnaireSum = 0;
    questionnaireSum += qResponse.getClassesValue();
    questionnaireSum += qResponse.getEatingValue();
    questionnaireSum += qResponse.getCampusVisitsValue();
    questionnaireSum += qResponse.getMaskWearingValue();
    questionnaireSum += qResponse.getSocialDistancingValue();
    questionnaireSum += qResponse.getTransportationValue();
    return questionnaireSum;

Function Get Contacts Value(SubjectID):
    cResponse = GET contactValue from Database with SubjectID;
    /* This get request takes the SubjectID as input and
       returns the sum of all contactValues that SujectID has
       within a 24 hour period as a response                */
    contactsValue = cResponse.getContactsValue();
    return contactsValue;

Function Get Lask Risk Score(SubjectID):
    rsResponse = GET last risk score from Database with SubjectID;
    /* This get request takes the SubjectID as input and
       returns the most recent risk score for that SujectID
       as a response                                        */
    lastRiskScore = rsResponse.getLastRiskScore();
    return lastRiskScore;

Function Normalize(RiskScore):
    normalizedRiskScore = minimaxNormalization(RiskScore);
    return normalizedRiskScore;

Function Calculate Risk Score(questionnaireValue,
 contactsValue):
    return Normalize(questionnaireValue, contactsValue);

Function Main():
    /* SubjectID is stored locally in each user's app once
       they log in                                          */
    questionnaireValue = Get Questionnaire Value(SubjectID);
    contactsValue = Get Contacts Value(SubjectID);
    lastRiskScore = Get Last Risk Score(SubjectID);
    newRiskScore = (Calculate Risk Score (questionnaireValue,
     contactsValue) + lastRiskScore) / 2 ;
```

4.2 Machine Learning for Proximity Detection

The machine learning model was trained using data from the MITRE Range-Angle structured dataset [35]. The MITRE Range-Angle structured dataset consists of a series of Bluetooth advertisements collected by smartphones according to the MITRE Structured Contact Tracing Protocol. The dataset was submitted as part of an effort to enhance contact tracing technology by the Private Automated Contact Tracing (PACT) project. PACT is a project whose mission is to “enhance contact tracing in pandemic response by designing exposure detection functions in personal digital communication devices that have maximal public health utility while preserving privacy” [59].

The dataset consisted of 69 sessions, each of which followed the MITRE Structured Contact Tracing Protocol. During each session, there were two testers: the beacon and the receiver. The beacon stays in a single position for the duration of the session and possesses a smartphone that sends BLE signals. The receiver uses a smartphone and the BlueProximity app to receive and record BLE advertisements at various distances and angles from the beacon. For each session, each tester chose a location for their smartphone to be held (choosing from shirt pocket, front pants pocket, back pants pocket, in purse/bag, or in hand) and a body orientation (sitting or standing). The session took place in one of the following types of environment: a small room, a medium-sized room, a large room, a hallway, or outdoors [47]. Figure 4.4 illustrates the data contained in each BLE advertisement of the dataset.

Advertisement Type	Variable	Description
Angle	Angle	The angle between smartphones.
Range	Range	The distance between smartphones.
Bluetooth	Receipt Timestamp	A timestamp representing when the BLE advertisement was received.
	Device ID	The ID of the device that received the advertisement.

	RSSI	The RSSI of the BLE advertisement.
	TX Power Level	The transmission power of the BLE advertisement.
	Advertisement Timestamp	A timestamp representing when the BLE advertisement was transmitted.

Figure 4.4: A table displaying the data contained in the MITRE Structured dataset.

The model's input consisted of features extracted from a series of RSSI readings collected within a short duration of time. To obtain training data from this dataset, we transformed a dataset of RSSI readings into a dataset of labelled sets of features. Then we split the transformed data set into training, testing, and validation sets in the ratio 60%/20%/20% respectively. This dataset's training set was used to train a regression model. After determining the type of model to use, we attempted to use overlapping windows and the addition of a simple moving average feature during training in order to decrease model error. After determining the amount of window overlap and the simple moving average window that led to the lowest amount of validation error, we used those parameters to create our final model.

4.3 Modules

4.3.1 Module Design Diagram

Figure 4.5 illustrates our planned software design by showing each module of Goatvid, its interactions with other modules, and its interactions with services such as API's or servers. The app begins with the Subject ID authentication module that checks that the entered Subject ID is valid and has been assigned to a user. It also locally stores the SubjectID and password on the device and marks the SubjectID as assigned in the database. Next, the Self-Reported Habits module allows the user to enter behavioral information about themselves which informs the Personal Risk Measurement Module. The values associated with the users answers to each question are summed up and later used in the risk score calculation. The Contact Tracing Module takes distance calculations from the Linear Regression Model and BLE Distance Estimation in order to calculate close contacts. This module then stores the close contacts on the database.

These close contacts along with the Self-Reported Behavior module are used in the Personal Risk Measurement module which produces a percentage risk score for the user. In addition, the Statistics module displays data from the WPI COVID stats dashboard, and the Message Center module allows messages to be sent to all users through a Push Notification Manager.

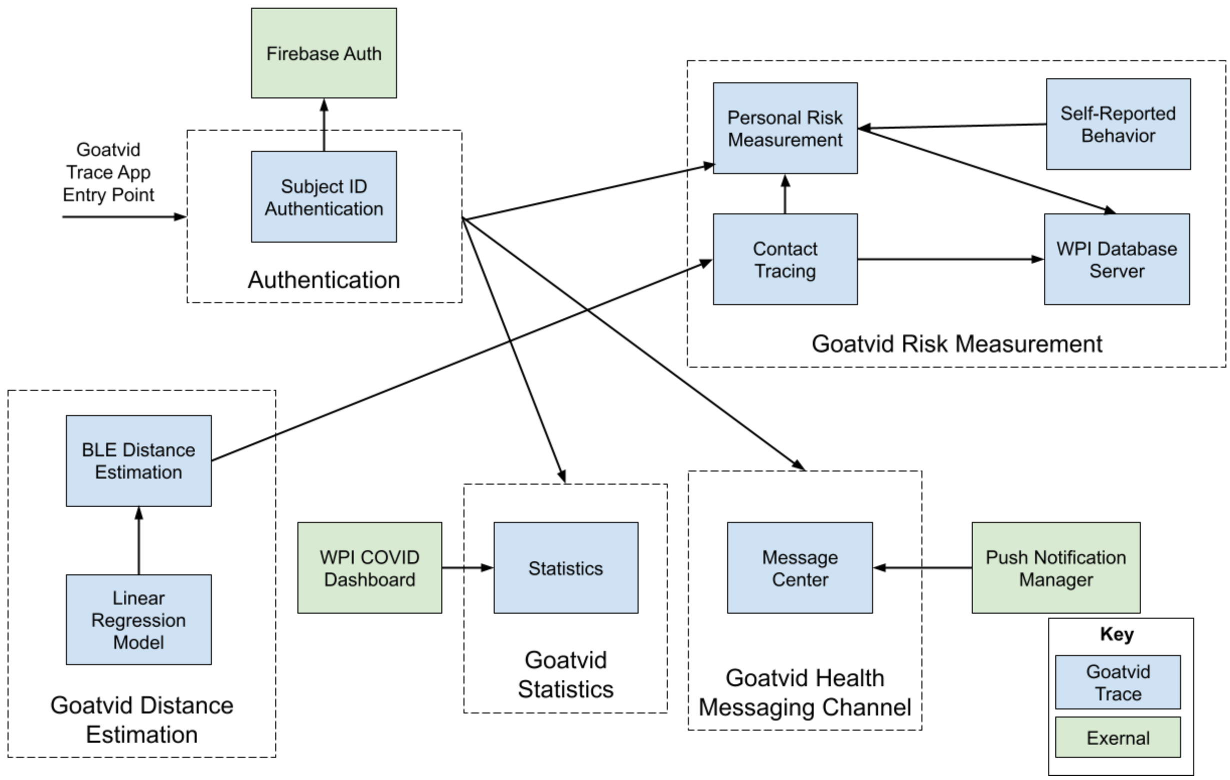


Figure 4.5: Goatvid Module Design Diagram

4.3.2 Personal Risk Measurement

This module calculates an estimated risk score that quantifies whether a user has been exposed to coronavirus. The data used for this calculation comes from both the Contact Tracing Module and the Self-Reported Behavior module. From contact tracing, we gain information such as number of contacts per day, number of close contacts, and average amount of time spent in close contact. We also record contacts with the types of places they were detected. Figure 4.6 shows the place types that are weighted by the app. The risk scores assigned to each location were based on the COVID-19 Risk Index infographic from COVID-19 RECoVERY CONSULTING [16]. This infographic which can be found in Appendix H, classified activities into 5 categories based on their risk of exposure to COVID-19. This information was combined with CDC guidelines and recommendations to reduce risk to create a numerical risk score [12].

		Place Type API Labels	Risk Scores
Location	Bar	bar	9
	Buffet	restaurant	9
	Music concert	night_club	9
	Movie theater	movie_theater	8
	Grocery store	supermarket	5
	Restaurant (outdoors)	restaurant	5
	Shopping (mall)	shopping_mall	5
	Small dinner party	*BLE advertising*	5
	Beach	natural_feature	5
	Swimming (public pool)	*establishment*	5
	Hair cut	*establishment*	5
	Take-out	restaurant	3
	Exercise (outside)	*use activity API*	2
	Pumping gas	gas_station	3
	Running, walking or biking with others	*use activity API*	2

Figure 4.6: A table displaying the weighting used for certain place types

This place type data is used to assign higher risk to someone who goes to a bar and a lower risk to someone who goes to a park, for example. Data from the Self-Reported Behavior module such as number of roommates and number of times a user goes grocery shopping a month is used in the calculation as well. This calculation is on a scale of 0 to 1 indicating the probability that a given user was exposed. Figure 4.7 shows the algorithmic flow of this module. The data collected in each part of the personal risk measurement process is shown in the bulleted lists below.

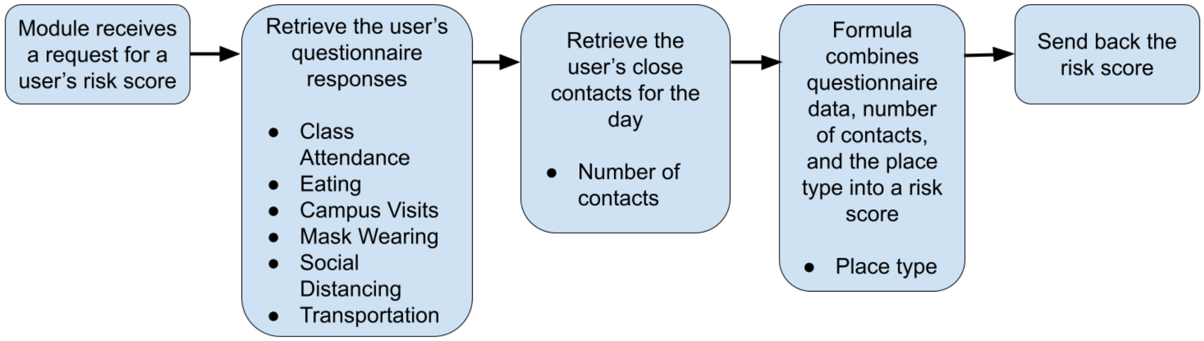


Figure 4.7: Personal Risk Measurement Flow Diagram

The formula is used to give each user a score between 0-100 to reflect how at risk they are of getting infected with the virus, where a score of 0 means the user is healthy and a score of 100 means the user is very likely to have caught COVID-19. The score is calculated by two functions using information provided from both a form and beacon library readings and stored in the database. The form collects personal information from the user with questions that require an integer as response and these values are then sent to the server. The questions cover personal information that we cannot measure from the user ranging from daily habits to number of roommates. The responses to these questions are weighted based on the table shown in Figure 4.8. In the server, the first formula sums up all the form values and times it by a multiplying value. This multiplying value is 1 plus the sum of each weighted contact daily. Contacts are measured using beacon library values to calculate distance and time stamps to measure time, a close contact is defined as being closer than 6ft for more than 15 minutes. The contact is then weighted by the value of the location score gained by the Places API. The second function then normalizes this result using a min-max normalization formula. Figures 4.8 and 4.9 show how the formula is calculated. The exposure is the starting point for determining one's risk score. It is then adjusted up or down using the Multiplying Values (MV) which is calculated from the number of close contacts they had and the locations they took place at. The resulting value is then normalized to create a digestible risk score percentage ranging from 0% to 100% to display to the user.

$$Risk\ Score = Normalization(Exposure * MV)$$

[43]

Figure 4.8: Risk Score Formula

Risk Score Breakdown		
Exposure		
The "Exposure" represents the sum of the values collected from the questionnaire listed below. We sum the values in order to treat each behavior from the questionnaire independently. This gives a holistic estimation of one's everyday risk by providing a sum of one's actions. Exposure = Class_attending_value + Food_source_value + Campus_visits_value + Mask_wearing_value + Social_distancing_value + Transportation_value		
Variable	Value type	Notes
Class_attending_value	Integer (0,5)	The higher the value the higher the risk. The range of values for each is chosen based on the number of options in the corresponding questionnaire question. For example, behaviors with more variation like transportation had more options and therefore a larger range of values.
Food_source_value	Integer (1,10)	
Campus_visits_value	Integer(1,5)	
Mask_wearing_value	Integer(1,5)	
Social_distancing_value	Integer(1,5)	
Transportation_value	Integer(0,7)	
Multiplying Value (MV)		

The “MV” represents the sum of the weighted close contacts plus 1

MV = 1 + c1 + c2 + … +cn (where cn is a value between 0 and 1)

Each contact is originally of value 1 but is then weighted by multiplying it by the tag of the places API type where it was measured. This is done to place a higher value on close contact interactions that take place at higher risk locations or during higher risk behaviors according to CDC guidelines [12]. This ensures that risky interactions such as going to a bar are given more weight in the final risk score than interactions that occur outside at a park. This ensures that risky interactions such as going to a bar are given more weight in the final risk score than interactions that occur outside at a park. For example:

Close contact = 1

Places API type(tag_value) = grocery store (9)

Weighted contact = C*(tag_value*10^{-1})

Normalization Factor

Minmax normalization is calculated on the result of the multiplication between “MV” and “Exposure”.

Risk Score Twice Daily Update

risk_score = (previous_risk_score + new_risk_score) / 2

Figure 4.9: Risk Score Variables breakdown

The tag_value for weighting the contact is provided by the places API type using these values in Figure 4.10:

Location	Place Type API Labels	Risk Scores
Bar	bar	9
Restaurant	restaurant	9
Night Club	night_club	9
Movie Theater	movie_theater	8
Supermarket	supermarket	5
Shopping Mall	shopping_mall	5
Natural Feature	natural_feature	5

Figure 4.10: Place type tag values

The form includes 6 multiple choice questions to gage the user's habits, the questions are presented using this format in Figure 4.11:

1. Which of the options below align most closely with the way in which you attend classes? (Course related reasons include lectures, labs, office hours, group meetings, project work, etc.)

    a. Never leave the house for course related reasons

    b. Going to campus < 1x a week for course related reasons

    c. Going to campus 1x a week for course related reasons

    d. Going to campus 2-3x a week for course related reasons

    e. Going to campus 4-5x a week for course related reasons

    f. Going to campus 6x a week or more for course related reasons

Figure 4.11: Questionnaire sample

To see the whole questionnaire, check Appendix A.

Each question from the questionnaire matches the response choices with a numeric value to attribute as shown in Figure 4.12:

Questionnaire Value relations						
Question 1: Class attending value	Never leave the house for course related reasons	Going to campus < 1x a week for course related reasons	Going to campus 1x a week for course related reasons	Going to campus 2-3x a week for course related reasons	Going to campus 4-5x a week for course related reasons	Going to campus 6x a week or more for course related reasons
Value	0	1	2	3	4	5

Figure 4.12: Questionnaire Values

To see the whole questionnaire responses value relations, check Appendix B.

4.3.3 Subject ID Authentication

To accurately track user's risk data, the app needs to associate each user with a username and password. This could be done using the users email and password. However, to provide authentication while still retaining the anonymity required for a user's study, the app will use randomly assigned SubjectIDs as usernames. This solution allows each user's data to be clearly recorded on the backend without their identity being exposed or associated as well as ensuring only registered participants can partake in our study.

4.3.4 Contact Tracing

For the app to be able to estimate a user's risk of contracting COVID19, it needs to keep track of contacts. Thus, this module is used to detect and record contacts. It uses the BLE Distance Estimation module to detect other smartphones. If another phone is closer than 6 feet for more than 15 minutes, it will be recorded as a close contact. This is called the Too Close for Too Long detection [64]. The module records close contacts by storing them in a remote database server located at WPI. This solution is accurate because it is based on the CDC's advice on close contacts. In addition, storing the contacts in a database enables our app to retrieve them so they can be used to calculate the risk score. Figure 4.13 shows the contact tracing process for our app.

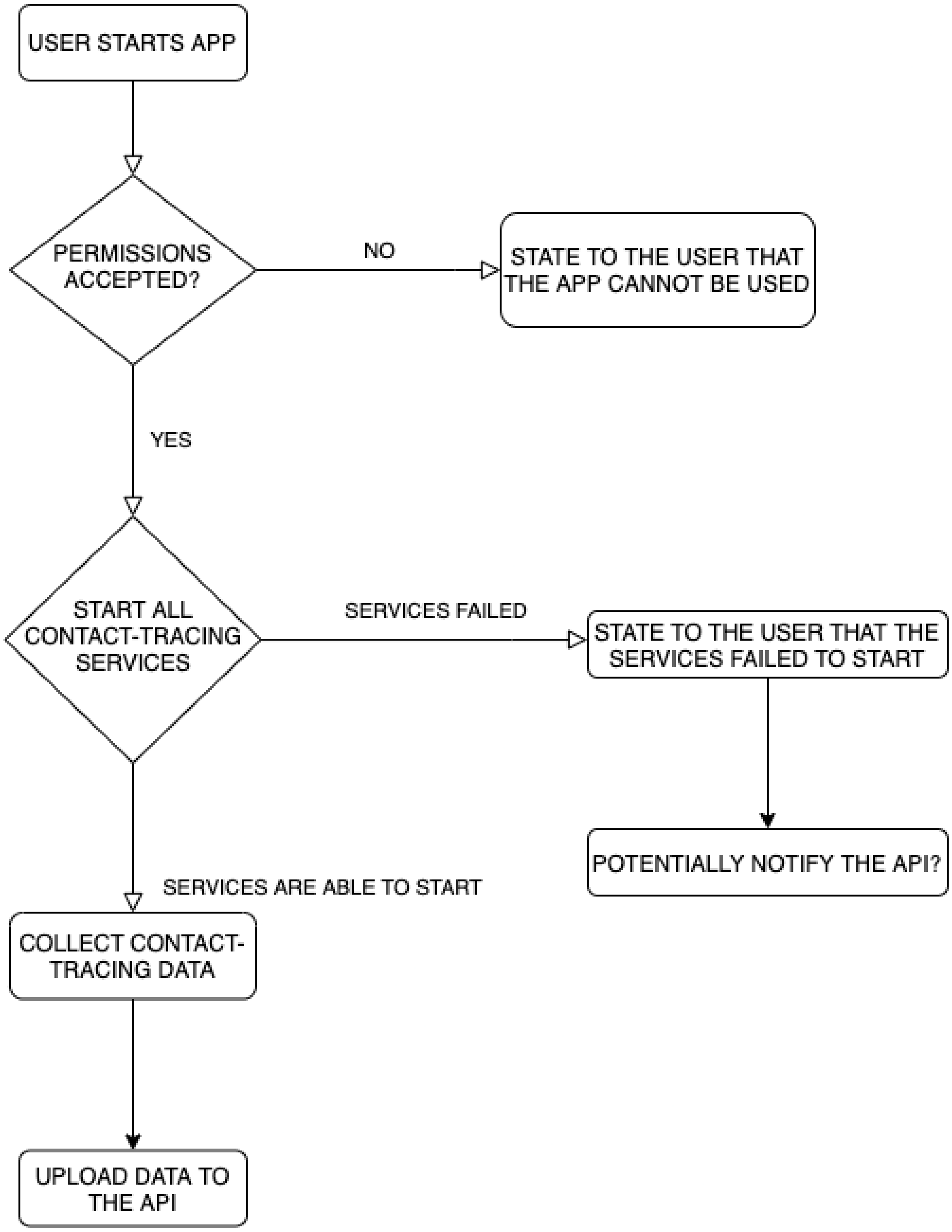


Figure 4.13: Contact Tracing Flow Diagram

4.3.5 Self-Reported Behavior

This module aims to obtain static data from the user. This includes data that we would be unable to obtain through contact tracing or other sensors. To obtain this data, there is a form integrated into the app. A user only needs to fill out the form once when they initially install the app, but they could fill it out multiple times if their situation has changed. The table below outlines the data collected in the form. The complete survey can be found in Appendix A. The infographics used to inform these reasonings can be found in Appendix H. Figure 4.14 shows the behaviors tracked by the questionnaire and the rationales for incorporating them.

Behavior	**Rationale for including this behavior**
Attending class	Online class is recommended by the CDC when available to reduce risk of exposure [10].
Getting Food	Food and grocery delivery is recommended by the CDC when possible to reduce risk of exposure [12].
Going to Campus	Increased personal contact increases one's risk of being exposed [10].
Mask Wearing	Wearing a mask limits exposure to COVID-19 particles [10].
Social Distancing	Close personal interaction increases one's risk of being exposed [10].
Use of Transportation	Transportation options (walking, rideshare, public transportation) have widely different risks of exposure [12].

Figure 4.14: Rationale for collecting data in the in-app form for Self-Reported behavior

4.3.6 BLE Distance Estimation

To enable the app to estimate distances between smartphones, it needs to be able to record BLE signals and use them to estimate the distance. That is the function of BLE distance estimation module. The BLE distance estimation module receives BLE signals from other smartphones, transforms them into an appropriate input format if the machine learning model is being used, and then uses the signals to carry out a distance estimation procedure. We decided to transform RSSI data for the machine learning model by formatting them into a time series, separating the time series into windows of time, then deriving features such as the mean RSSI from the windows. Figure 4.15 illustrates the process of transforming the BLE signal data.

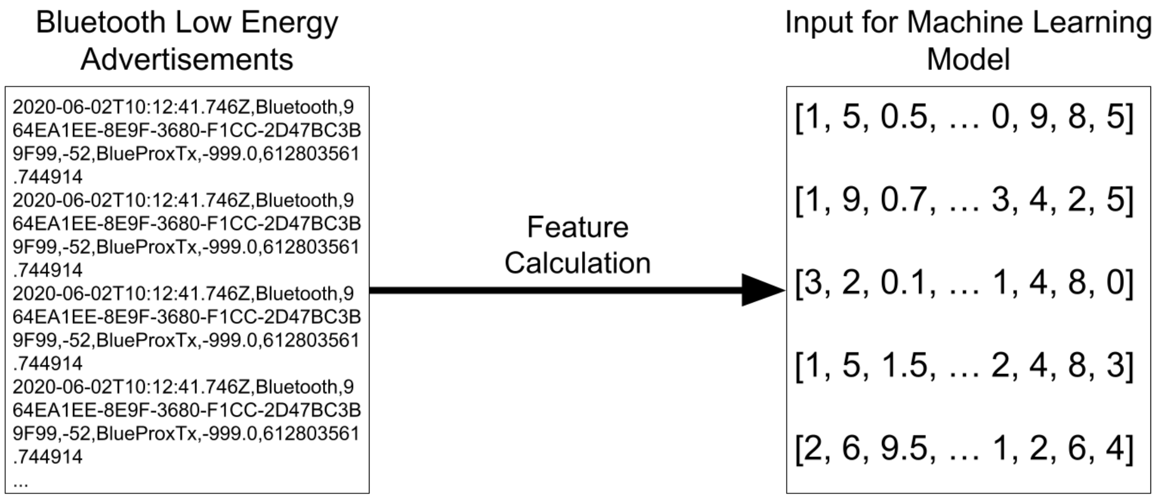


Figure 4.15: BLE RSSI Input Transformation

The distance estimation procedure would be either to use the AltBeacon library or to send the input to the machine learning model to retrieve a distance prediction. This module makes it easier to conduct tests on our app because we can easily edit this module to carry out an arbitrary distance prediction method. As a result, it quickens the process of testing the app's distance prediction methods. Figure 4.16 shows the algorithmic flow of this module.

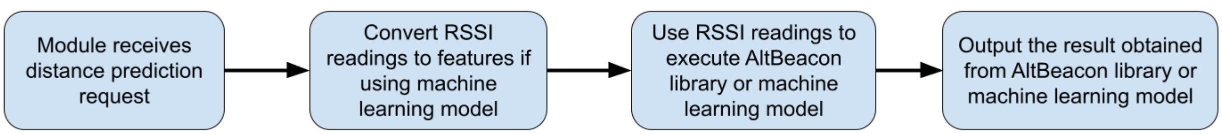


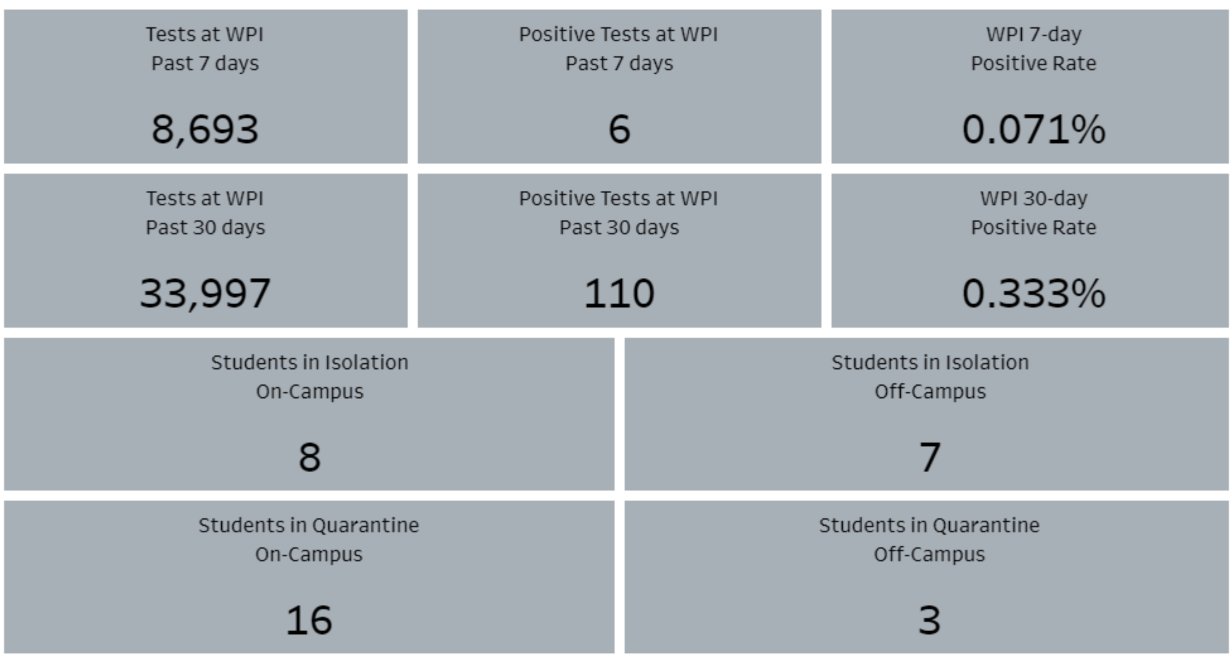
Figure 4.16: BLE Distance Estimation Flow Diagram

4.3.7 Health Services Communication Channel

Emerging information on factors or statistics that impact COVID transmission and risk scores needs to be disseminated to the WPI community. Currently, these updates are typically communicated through email. One example of how we imagine this Health Services Communication Channel could be used is when WPI made the announcement that neck gaiters are no longer considered valid face coverings. If a mobile application such as this one existed, WPI Health Services could have used it to make the announcement. We also implemented a 'message center' or inbox type feature that stores a history of all these push notifications. This inbox also allows for push notifications to be clicked and opened to display more in-depth articles or redirect to an external webpage. This module serves as a way for Health Services to quickly communicate COVID and other health related information to the whole WPI student body.

4.3.8 Statistics

In order to keep users up to date with the COVID statistics in their area, they need a page with recent statistics. Depending on which community the app serves, the source of the statistics would be different. This module displays statistics pertaining to both the WPI community and the larger Worcester and Massachusetts community because the app is aimed for WPI students. The data for this module is taken from the WPI's COVID Testing Dashboard which is updated daily [70]. It shows statistics such as the number of positive cases and positive test rate for WPI and Worcester. These statistics are compiled on the UI for the user to know the status of COVID in the WPI community. This allows users to easily find all relevant COVID information in one app. Figure 4.17 shows the WPI Dashboard and Figure 4.18 displayed the associated definition for each statistic referenced within our app .



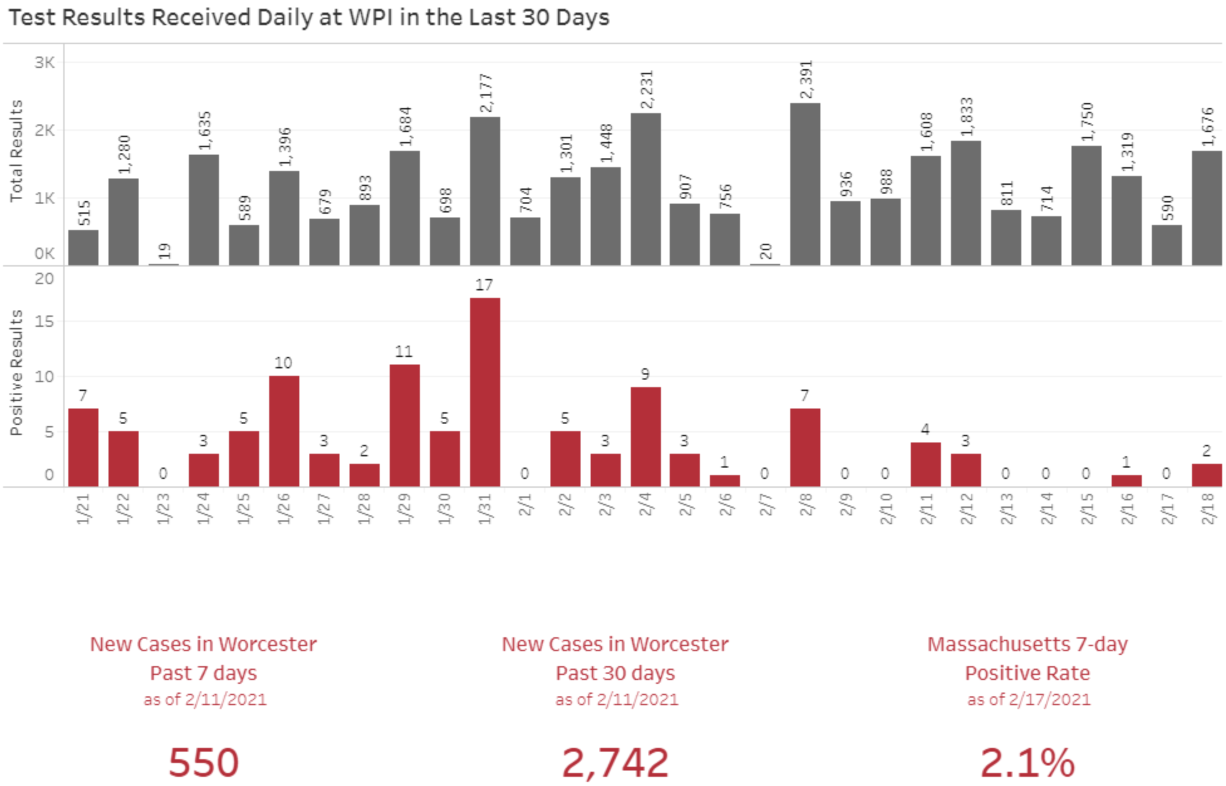


Figure 4.17: Example Data from WPI's COVID Dashboard

Statistic	**Definition**
Positive Tests at WPI Past 7 Days	The rolling total of all positive test results received in the past 7 days.
Positive Tests at WPI Past 30 Days	The rolling total of all positive test results received in the past 30 days.
WPI 7-day Positive Rate	The percentage of all valid test results received in the past 7 days with a positive test result.
WPI 30-day Positive Rate	The percentage of all valid test results received in the past 30 days with a positive test result.
Tests performed at WPI Past 7 Days	The rolling total of all tests results received in the past 7 days.
Tests performed at WPI Past 30 Days	The rolling total of all test results received in the past 30 days.
Massachusetts 7-day Positive Rate	The percentage of all reported tests in the state with a

	positive test result.
New Cases in Worcester Past 7 days	The rolling total of all new cases reported in the City of Worcester in the past seven days.
New Cases in Worcester Past 30 days	The rolling total of all new cases reported in the City of Worcester in the past 30 days.

Figure 4.18: Definition of WPI Dashboard Stats

4.3.9 Machine Learning Model

For the app to be able to detect close contacts, it needs a method of estimating distances between smartphones. The machine learning model does this by using BLE signals as input to predict distances between smartphones. Another approach could be to use a library such as AltBeacon to estimate distances. This approach is useful because using machine learning to estimate distances from BLE signals is an approach that can be implemented more quickly than other approaches such as manually processing BLE signals. An illustration of this process is shown in Figure 4.19.

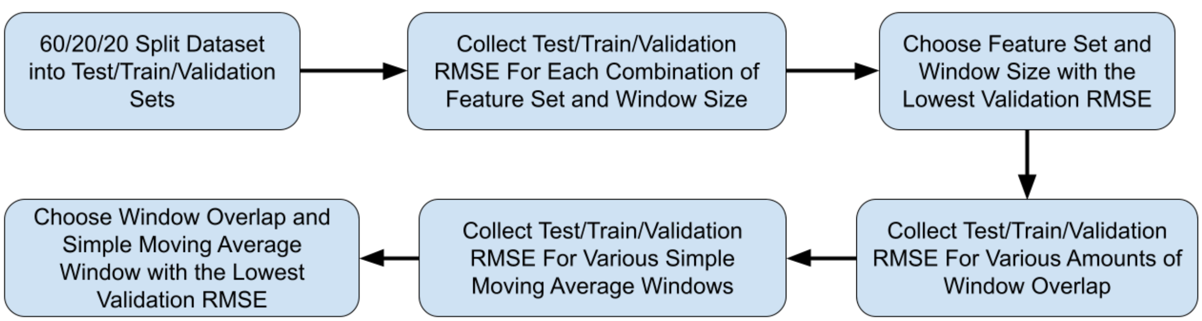


Figure 4.19: Machine Learning Model Flow Diagram

We hosted the machine learning model remotely so that the application does not require too much computation to predict distance between smartphones. We consider this a priority given that the application would be constantly predicting distances in an ideal setting.

4.3 User Interface Flow Diagram

The basic user interface of this app is tabbed. This allows the user to easily navigate between functionalities of the app. Since this is an app that students would be encouraged to download by WPI, it is important that it is straightforward and easily usable to avoid frustration.

The first page is an authenticated login page. Once logged in, the user is presented with a 2 page tabbed app. The first tab displays a user's current risk score. The second tab contains WPI and Worcester COVID statistics. The Risk Score page has buttons to allow the user to navigate to the Self-Reported Behavior form and the Health Services Message Center. The flow of our app's user interface is illustrated in Figure 4.20.

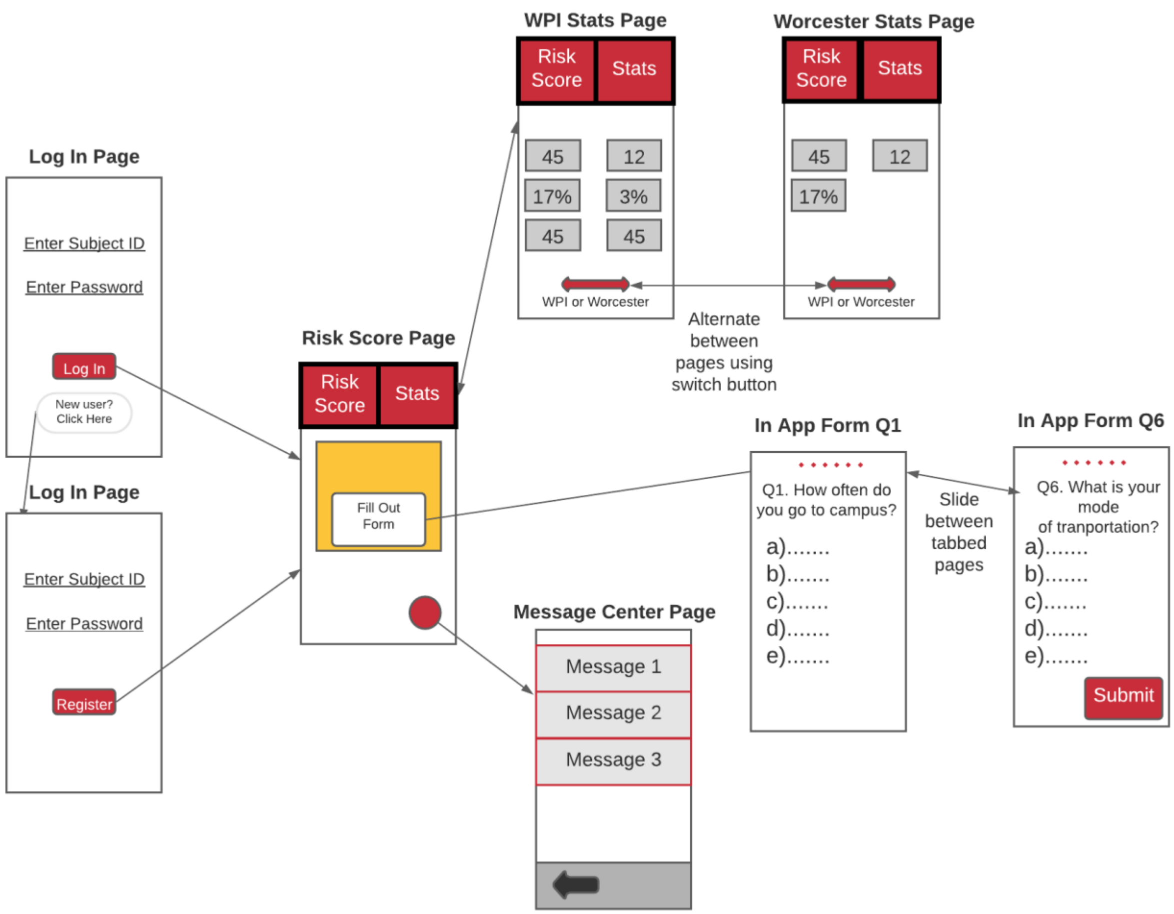


Figure 4.20: Goatvid Trace Mobile App User Interface Flow Diagram

Chapter 5. Implementation

Our implementation of the design was an Android mobile application created in Android Studio. Specifically, the compile and target SDK version we programmed for was version 29 with a minimum SDK version of 21. When testing our application, we used the Moto G5 Plus smartphone.

5.1 System Architecture Diagram

The app is based on activities and the ViewModel class. The ViewModel is a UI wrapper to manage UI-related data in a lifecycle conscious way. The app also uses a repository to pull data from. This repository includes a database model and a remote data source. The database model is created using the Room API and connects to an SQLite database. The remote data source uses an HTTP API to pull data from both our web service and the WPI COVID dashboard. Figure 5.1 shows this architecture and the connections between its components.

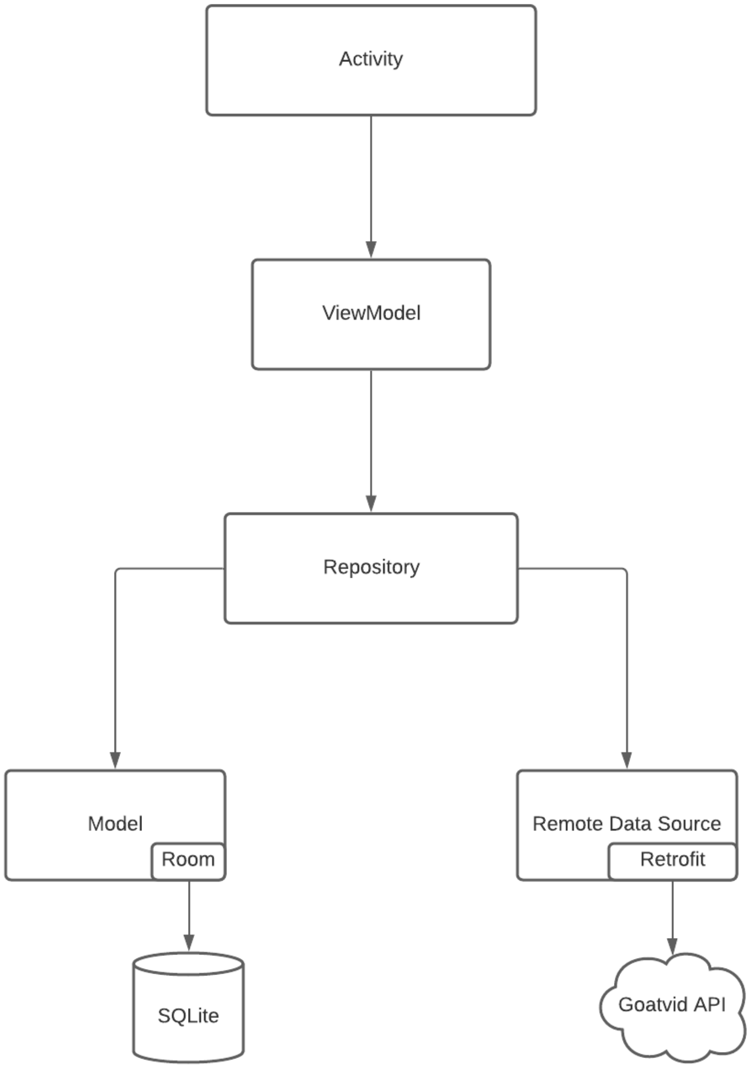


Figure 5.1: System Architecture Diagram

5.2 Risk Score Formula

The formula was coded in java in the front end and from there was connected to the database through endpoints. The formula runs twice a day using the AlarmManager class.

The Formula.java class is doing all the necessary calculations and it is included in the model as part of the Repository.

First, we have the alarm manager which in turn calls the update function:

```java
* This function sets the alarm and creates the broadcast receiver that runs when the alarm
* runs, which in turn sets the next alarm. It also has permissions set to schedule an alarm
* whenever the device restarts.
*/
static public void setAlarm()
{
    String subjectID = RegistrationActivity.credentials.getSubjectID();
    int id = Integer.valueOf(subjectID);
    BroadcastReceiver receiver = new BroadcastReceiver() {
        @Override public void onReceive(Context context, Intent _ )
        {
            flag = false;
            dayPassed = true;
            //updateRiskScore();
            getQuestionnaireValue(id);
            context.unregisterReceiver( this );
            System.out.println("$$$$$$$$$ ALARM HAS BEEN RUN");
        }
    };

    flag = true;
    getApplicationContext().registerReceiver( receiver, new IntentFilter( action: "update formula value") );

    PendingIntent pintent = PendingIntent.getBroadcast( getApplicationContext(),  requestCode: 0,
            new Intent( action: "update formula value"),  flags: 0 );
    AlarmManager manager = (AlarmManager)(getApplicationContext().getSystemService( Context.ALARM_SERVICE ));

    // set alarm to fire in 24 hours (1000*60*60*24) from now (SystemClock.elapsedRealtime())
    manager.set( AlarmManager.ELAPSED_REALTIME_WAKEUP,  triggerAtMillis: SystemClock.elapsedRealtime() + 1000*60*60*24, pintent );
```

The update function includes this if statement that calls the calculating function:

```java
/*
* This if statement calls the calcRS which calculates the risk score and implements
* the rolling average when appropriate before updating the variable.
*/
if(respArray[2] == -1){
    double temp = 2.0 + respArray[1];
    newRS = (calcRS(respArray[0], temp));
    System.out.println("MV " + temp);
    System.out.println("QUESTIONNAIRE" + respArray[0]);
    System.out.println("NOT AVERAGED RISK SCORE AMOUNT " + newRS);
} else {
    double temp = 1.0 + respArray[1];
    newRS = (calcRS(respArray[0], temp) + respArray[2]) / 2;
    System.out.println("AVERAGED RISK SCORE AMOUNT " + newRS);
}
```

The calculating function returns the risk score by calling the normalization function on the calculated value:

```java
/*
* This function is called by the update function and calculates the current risk score
* addi = the sum of the questionnaire response values; provided by the server
* mv = sum of the close contacts recorded in the last 24 hours; provided by the server
*
* returns: the risk score before the rolling average
*/
public static double calcRS(double addi, double mv){
    double q = addi;
    System.out.println("The value of the  q " + q + "\n\n\n");
    double mval = mv;
    System.out.println("The value of the  mval " + mval + "\n\n\n");
    double mul = q * mval;
    System.out.println("The value of the  mul " + mul + "\n\n\n");
    double resu = normalizeSTAD(mul);
    System.out.println("The value of the  resu " + resu + "\n\n\n");
    return resu;
}
```

The normalization function runs a minimax normalization on the value it takes as a parameter:

```java
/*
* This function is called by calcRS and it runs minmax normalization on the input
* x = value to normalize
*
* return: Normalized value
*/
public static double normalizeSTAD(double x){
    double vton = x;
    System.out.println("The value of the  vton " + vton + "\n\n\n");
    double rest = vton -4.0;
    System.out.println("The value of the  resta " + rest + "\n\n\n");
    double div = rest / 87.0;
    System.out.println("The value of the  div " + div + "\n\n\n");
    double res = div * 100;
    System.out.println("The value of the  resNOrm " + res + "\n\n\n");
    return res;
}
```

5.3 Push Notifications

Push notifications are used in our app as a channel for WPI's HealthServices to quickly communicate with the WPI community about COVID related updates. Specifically, we used Airship, a user engagement platform for mobile devices. Airship provides push notification services, in-app messaging, as well as an inbox style message center. We followed the Airship 'Getting Started' page and used Airship SDK 14.0.0. We chose FCM as our Push Notification Provider [2].

5.3.1 Using Firebase Cloud Messaging

In order to communicate with users about COVID related announcements, we began implementing push notifications using Firebase Cloud Messaging (FCM). We were able to easily integrate FCM with our app and begin drafting and sending out push notifications from the online console almost immediately. However, FCM did not come with a pre-made inbox UI. This required all notifications to be created programmatically in order to achieve our desired experience. We want WPI administrators from Health Services to quickly and easily send out notifications. Requiring the notifications to be created programmatically does not achieve this.

5.3.2 Using Airship Customer Engagement Platform

In order to provide an effective channel for WPI administrators to communicate COVID related announcements, we decided to switch to the Airship Customer Engagement Platform for push notifications. Airship for android used FCM as a backend and was very easy to install following the guide online. We installed Airship version 14.0.0 for our implementation. Like FCM, Airship has a console type website called a dashboard that allows someone to craft their notification online before sending it. Airship also comes with a pre-build inbox type page called the Message Center. Together, this allows a push notification to open directly to the Message Center on click as we intended. Additionally, the Airship Dashboard has lots of options for designing push notifications and the pages the pages that they link to. This allows for a lot of flexibility for how WPI admin can choose to communicate their message. The figures below show the process a WPI administrator might take to create their message in the Airship Dashboard. Figures 5.2 and 5.3 shows the Airship Dashboard page where the push notification content can be configured. This page also allows the message creator to link their push notification to the in-app Message Center [2].

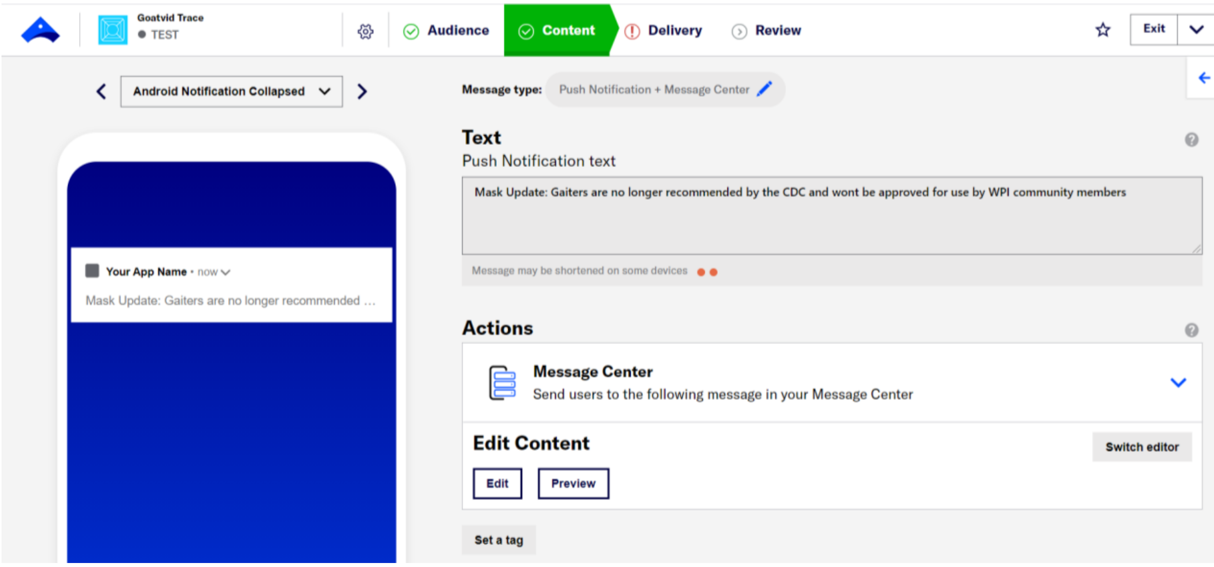


Figure 5.2: The main Airship Dashboard page used to create push notifications (Airship)

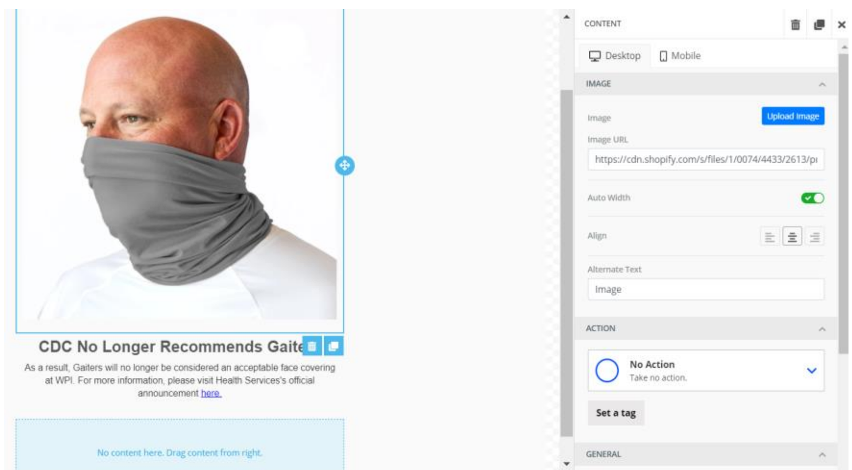


Figure 5.3: Screenshot of customizing a message through the Airship Dashboard (Airship)

Figure 5.4 demonstrates the use case of using Airship to send out information about the Gaiters update. From left to right, the images show the flow that the user can follow. The first image shows the users mobile device receiving a push notification from their Goatvid Trace app. On click, they are taken to the second image, a multimedia page providing more information about the update. The user can click on the link which automatically navigates to WPI's official announcement in their web browser as seen in the third image. Finally, the last image shows how this message is saved in the user's message center inbox so they can refer to it later on (Airship).

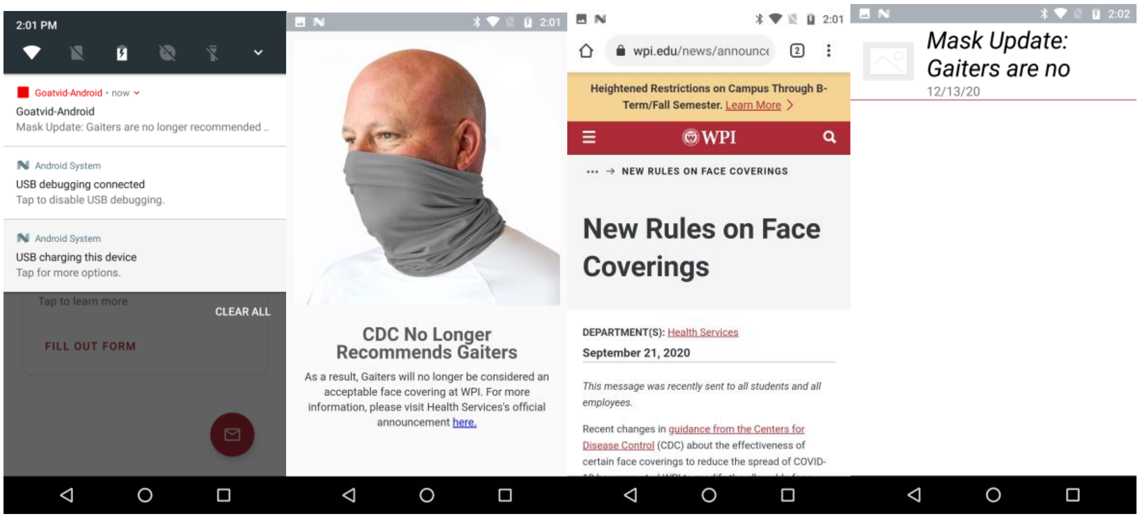


Figure 5.4: Flow chart of a user receiving a push notification from the Airship console (Airship)

5.4 Server

We created a WPI hosted server for this project. The server runs Ubuntu 18.04.5 LTS. It holds our database, HTTP endpoints, and other scripts needed for the app. The server also offers an API used by the app to carry out distance estimation, contact tracing, and retrieving local COVID statistics. A client-server diagram is shown in Figure 5.5.

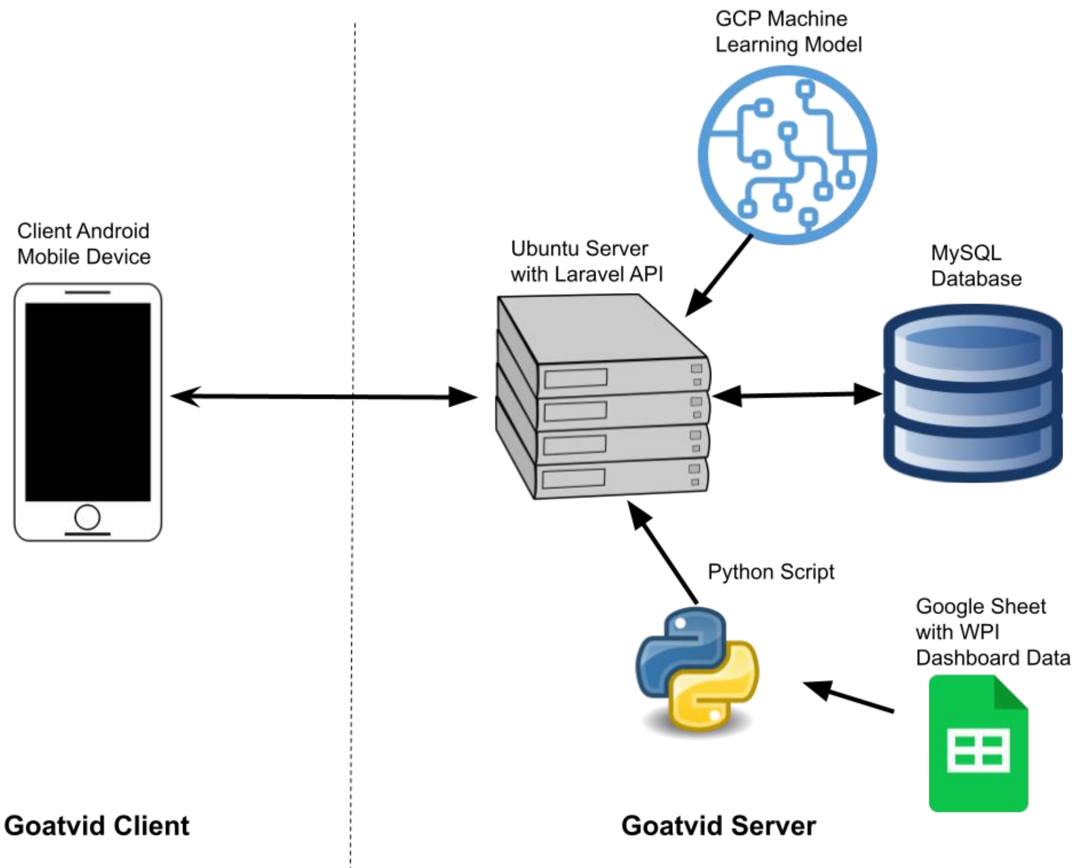


Figure 5.5: Client Server Diagram

5.4.1 Database

We decided to use MySQL as our main database to store close contacts, questionnaire values, subject IDs, risk scores, and WPI COVID dashboard statistics. The database consists of a few tables (contacts, wpi_stats) in which we are storing contact tracing data collected from the mobile application and other data such as the most recent number of cases at WPI. We are also using a PHP-based framework that supports writing database models [37] and object-relational mapping (ORM). Each database table has a corresponding model which is used to interact with it. In order to write queries, a built-in query builder [36] is used for convenience. The built-in query builder also provides protection against SQL injection attacks by default. Figure 5.6 shows the schema for our database.

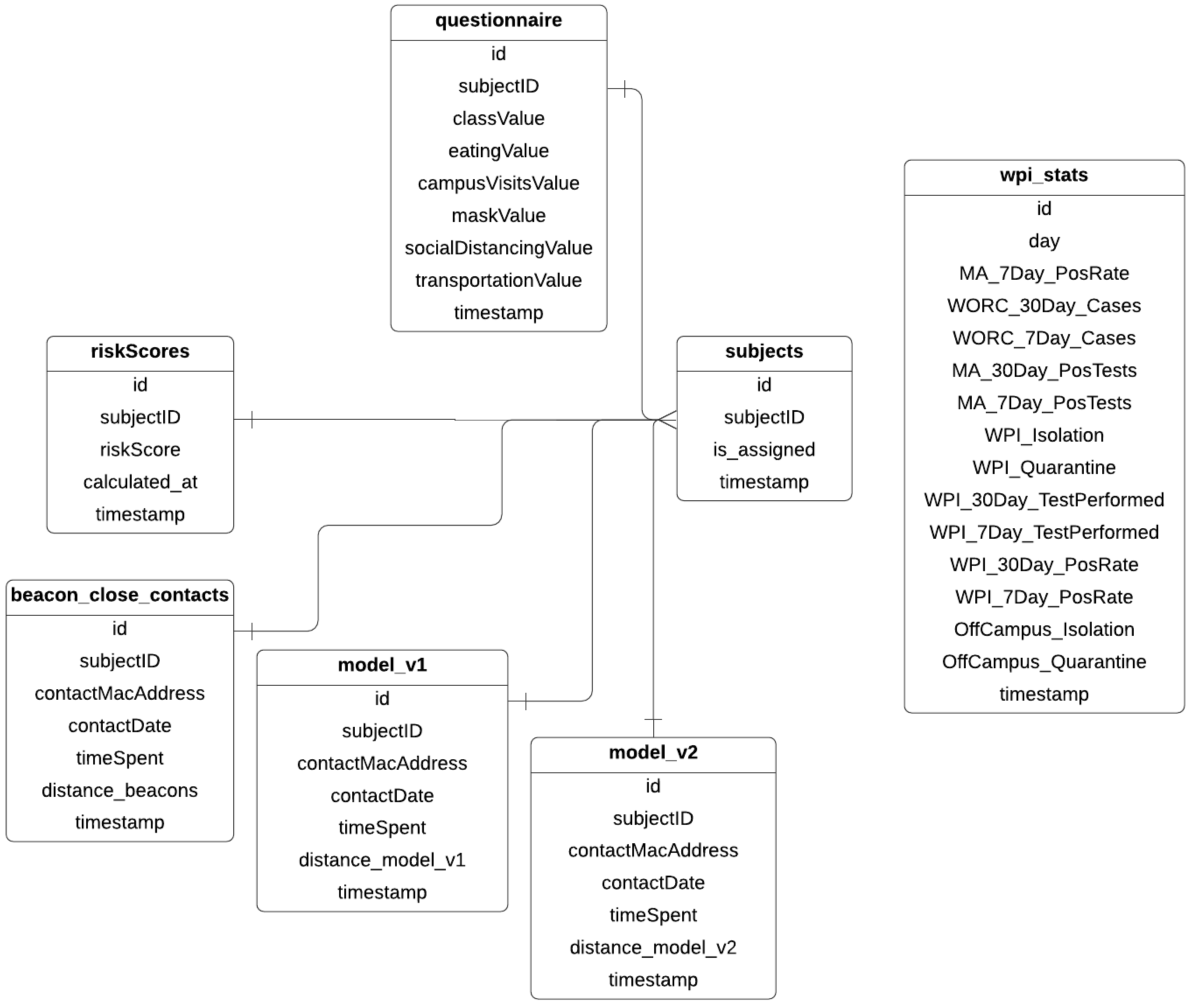


Figure 5.6: Database schema

We stored the SubjectIDs in a table with the following attributes. Figure 5.7 illustrates this table:

Attribute Name	**Type**	**Description**
id	integer	ID of entry within the table
subject_id	integer	Subject ID of the user
is_assigned	boolean	If the subject_id has already been assigned during registration
timestamp	datetime	Time at which the entry was updated

Figure 5.7: Attributes in subject_users table

The results from the in-app questionnaire [Appendix A] are stored in a table with the following attributes. Figure 5.8 illustrates this table:

Attribute Name	**Type**	**Description**
id	integer	ID of entry within the table
subjectID	integer	SubjectID of the user
classValue	integer	Integer 1-10 according to user's response to question 1
eatingValue	integer	Integer 0-5 according to user's response to question 2
campusVisitsValue	integer	Integer 1-5 according to user's response to question 3
maskValue	integer	Integer 1-5 according to user's response to question 4
socialDistancingValue	integer	Integer 1-5 according to user's response to question 5
transportationValue	integer	Integer 1-11 according to user's response to question 6
timestamp	datetime	Timestamp when questionnaire entry was submitted to database

Figure 5.8: Attributes in questionnaire table

For the beacon_close_contacts table, we used the following attributes. Figure 5.9 illustrates this table:

Attribute Name	**Type**	**Description**
id	integer	ID of entry within the table
subjectID	integer	Subject ID of the user
contactMacAddress	string	MacAddress of device that the user made contact with
contactDate	date	Date that the contact was made

timeSpent	integer	Duration of time where devices are considered in close contact of each other
distance_beacon	double	Distanced estimated between the two users using beacon library
timestamp	datetime	Timestamp when initial contact was made

Figure 5.9: Attributes in beacon_close_contacts table

For the modelv1_close_contacts table, we used the following attributes. Figure 5.10 shows this table:

Attribute Name	**Type**	**Description**
id	integer	ID of entry within the table
subjectID	integer	Subject ID of the user
contactMacAddress	string	MacAddress of device that the user made contact with
contactDate	date	Date that the contact was made
timeSpent	integer	Duration of time where devices are considered in close contact of each other
distance_model_v1	double	Distanced estimated between the two users using machine learning model 1
timestamp	datetime	Timestamp when initial contact was made

Figure 5.10: Attributes in modelv1_ close_contacts table

For the modelv2_close_contacts table, we used the following attributes. Figure 5.11 shows this table:

Attribute Name	**Type**	**Description**
id	integer	ID of entry within the table
subjectID	integer	Subject ID of the user
contactMacAddress	string	MacAddress of device that the user made contact with

contactDate	date	Date that the contact was made
timeSpent	integer	Duration of time where devices are considered in close contact of each other
distance_model_v1	double	Distanced estimated between the two users using machine learning model 2
timestamp	datetime	Timestamp when initial contact was made

Figure 5.11: Attributes in modelv2_close_contacts table

For the wpi_stats table, we used the following attributes. Figure 5.12 illustrates this table:

Attribute Name	**Type**	**Description**
id	integer	ID of entry within the table
day	date	Date statistic entry was created
MA_7Day_PosRate	double	The percentage of all reported tests in the state with a positive test result.
WORC_30Day_Cases	integer	The rolling total of all new cases reported in the City of Worcester in the past 30 days.
WORC_7Day_Cases	integer	The rolling total of all new cases reported in the City of Worcester in the past seven days.
WPI_30Day_PosTests	integer	The rolling total of all positive test results received in the past 30 days.
WPI_7Day_PosTests	integer	The rolling total of all positive test results received in the past 7 days.
WPI_Isolation	integer	The number of students currently in WPI's dedicated isolation space.
WPI_Quarantine	integer	The number of students currently quarantining in residence halls, fraternities and sororities, or WPI dedicated quarantine space.
WPI_30Day_TestPerformed	integer	The rolling total of all test results received in the past 30 days.
WPI_7Day_TestPerformed	integer	The rolling total of all tests results received in the

		past 7 days.
WPI_30Day_PosRate	double	The rolling total of all positive test results received in the past 30 days.
WPI_7Day_PosRate	double	The percentage of all valid test results received in the past 7 days with a positive test result.
OffCampus_Isolation	integer	The number of students isolating in their off-campus apartments/homes, and those who have returned to their permanent residence.
OffCampus_Quarantine	integer	The number of students currently quarantining in their off-campus apartments/homes, and those that have returned to their permanent residence.
timestamp	datetime	Timestamp when stats entry was submitted to database

Figure 5.12: Attributes in wpi_stats table [70]

5.4.2 COVID Dashboard Data Processing

The WPI COVID Dashboard is a website hosted by WPI that displays COVID related statistics for the WPI community. The data is a combination of both WPI, Worcester, and Massachusetts COVID statistics and is updated at 4pm on weekdays. Unfortunately, the data on this dashboard is held within a Tableau Frame and we are unable to either download or scrape the data using Javascript.

As a temporary solution for this MQP, we accessed data from an unofficial google sheet created by the user u/ollien on the WPI subreddit. The first tab of the google sheet, seen in Figure 5.13 below, is an exact copy of the dashboard data and is updated once a day [66]. Unfortunately, this google sheet was only collecting data between 8/26/20 and 12/10/20. As a result, the mobile app was only able to show statistics as of 12/10/20.

Date Fetched	Massachusetts 7-day Positive Rate	New Cases in Worcester Past 30 days	New Cases in Worcester Past 7 days	Positive Tests at WPI Past 30 days	Positive Tests at WPI Past 7 days	Students in Isolation at WPI	Students in Quarantine at WPI	Tests Performed at WPI Past 30 days	Tests Performed at WPI Past 7 days	WPI 30-day Positive Rate	WPI 7-day Positive Rate	Students In Isolation Off-Campus	Students In Quarantine Off-Campus
10/27/2020 22:00:00	1.60%	610	191	6	1	0	7	32004	7948	0.02%	0.01%	4	5
10/29/2020 13:48:00	1.80%	610	191	7	2	0	6	32939	8054	0.02%	0.03%	2	5
10/30/2020 18:00:00	1.90%	705	236	7	1	0	6	33066	6699	0.02%	0.01%	0	5
11/2/2020 15:02:00	1.80%	705	236	10	4	1	2	30852	7849	0.03%	0.05%	3	2
11/3/2020 22:07:00	1.80%	705	236	10	4	1	0	31670	7184	0.03%	0.06%	6	2
11/4/2020 21:25:00	1.80%	705	236	10	4	4	0	32071	7391	0.03%	0.06%	7	2
11/5/2020 23:56:00	1.90%	705	236	10	3	4	0	33248	7710	0.03%	0.04%	2	1
11/9/2020 18:00:00	2.30%	830	266	13	5	2	18	29982	6556	0.04%	0.08%	8	13
11/10/2020 16:23:00	2.30%	830	266	15	7	6	19	31122	6949	0.05%	0.10%	5	9
11/11/2020 14:35:00	2.60%	830	266	15	7	8	21	32252	7532	0.05%	0.09%	8	10
11/12/2020 15:30:00	2.90%	830	266	19	11	8	21	32120	7190	0.06%	0.16%	13	14
11/13/2020 21:41:00	2.90%	1211	518	18	10	8	23	32549	7555	0.06%	0.13%	10	14
11/16/2020 14:54:00	3.10%	1211	518	24	14	11	20	30226	7864	0.08%	0.18%	6	23
11/17/2020 16:35:00	3.20%	1211	518	34	22	11	38	31076	7503	0.11%	0.30%	13	64
11/18/2020 17:16:00	3.20%	1211	518	38	26	11	39	32263	7596	0.12%	0.35%	13	65
11/19/2020 22:49:00	3.30%	1211	518	44	28	17	48	32653	7658	0.14%	0.37%	18	95
11/20/2020 16:42:00	3.20%	1687	667	44	28	16	49	31931	6221	0.14%	0.46%	18	94
11/23/2020 18:00:00	3.20%	1687	667	52	29	18	44	30640	7777	0.17%	0.38%	17	95
11/24/2020 18:00:00	3.10%	1687	667	53	20	19	39	31373	7658	0.17%	0.27%	16	95
11/25/2020 15:40:00	3.00%	1687	667	56	19	18	37	32875	8041	0.17%	0.24%	20	96
11/30/2020 18:00:00	3.90%	2001	550	58	8	9	11	28358	5161	0.21%	0.16%	4	61
12/1/2020 17:58:00	3.90%	2001	550	61	10	12	17	28946	5015	0.22%	0.20%	6	33
12/2/2020 18:00:00	3.90%	2001	550	60	8	7	14	29580	4518	0.21%	0.18%	5	27
12/3/2020 15:41:00	4.90%	2001	550	63	7	4	7	29758	3320	0.22%	0.21%	7	12
12/4/2020 18:00:00	5.30%	2747	1012	63	7	4	7	28758	3413	0.22%	0.21%	9	16
12/7/2020 18:00:00	5.30%	2747	1012	63	11	6	12	25738	4887	0.25%	0.23%	7	21
12/8/2020 18:00:00	5.30%	2747	1012	62	8	5	9	25731	4319	0.25%	0.19%	11	20
12/9/2020 15:39:00	5.80%	2747	1012	70	15	5	15	27596	4538	0.26%	0.34%	11	24
12/10/2020 17:22:00	5.90%	2747	1012	70	14	5	14	26931	4087	0.27%	0.35%	12	22

Temporarily down for maintenence, given the dashboard update. Updates may be delayed.

Figure 5.13: Raw Data from the COVID Dashboard stored on u/ollien's Google Sheet

To get the data from this google sheet, our python script uses the GoogleSheets API which directly accesses the live data on the sheet. The script is run daily by a CRON job on the server and the data is used to update our database. This database is queried by the app to get the most recent set of data to be displayed on the Stats Tab. The specific libraries used in the script can be found in Appendix G.

5.4.3 Goatvid Server

The API is written in a PHP framework called Laravel (Version 8.36.2 https://laravel.com/docs/8.x). This framework is providing scaffolding which makes it easier to add controllers and write components. We decided to use it as it is a framework that favors convention over configuration making the development of our REST API more efficient. The REST API that we built communicates using HTTP verbs as actions (GET, POST, PUT, DELETE). Requests can be performed by the mobile application which uses the API's endpoints to retrieve, create, update and delete resources.

5.5 Machine Learning Model

We created our machine learning model using Python. We chose to use Python's Scikit-learn library to facilitate machine learning, since its features shorten the amount of time needed to carry out tasks related to machine learning. The versions of Python and Scikit-learn used for machine learning were 3.7.3 and 0.23.2 respectively. To train our model, we used the MITRE Range-Angle structured dataset. It consisted of Bluetooth Low Energy advertisements transmitted between smartphones at distances between 3 to 15 feet and at different angles. The smartphones were also at different orientations. We conducted tests using the following model types provided by Scikit-learn: Lasso Linear Regressor, Random Forests, LinearSVR, K Nearest Neighbors, and XGBoost.

5.5.1 Deployment and Hosting

To deploy our final model to a binary file, we used Python's pickle library. We used Google Cloud Platform's AI Platform service to remotely deploy and host the machine learning model. This was done firstly by uploading the model file to a Google Cloud storage bucket. Then, we used Google AI Platform to remotely host the uploaded model. This allowed the application to request distance predictions by sending HTTP requests to a Google Cloud Platform API.

5.6 Subject ID Authentication

The user authentication process begins when a user creates a signs up for an account in the app. They enter a Subject ID number and password to register. The registration is valid if the Subject ID number entered by the user is in the list of assigned subject IDs in the database and has not already been registered. An API endpoint checks for these two things the user can create his/her profile.

5.7 Beacon Library

In addition to the various methods we used for contact tracing, we discovered that there is a library called AltBeacon (version 2.17.1) [54] which can be used to do distance measuring out of the box. We used the AltBeacon library to estimate distances between phones in order to calculate close contacts. A beacon is a “packet” containing unique identifiers. One of those identifiers is RSSI which is then used to do distance measurement. According to the library, 97 percent of the devices currently support this functionality. We integrated this library into the mobile application and set up an Android foreground service in order to run its beacon broadcasting and detection functions in the background. We had to use a foreground service to keep the service running in the background since we are working with newer Android versions (Android 10).

Using RSSI values, the library is able to provide estimates for beacon distances in meters. For example, when the two devices are about 1 meter the estimated distance is between 0.5 and 2 meters. As the distance between the devices is increased, the measurements tend to be less accurate because of the noise on the signal measurement. [55]

We configured the library to be used within an Android foreground service. The difference between a regular service is that a notification is constantly appearing while the service is running. The reasoning is that devices that run Android 8 or a greater version are restricting the services’ runtime to only 10 minutes. [56]

According to a blog post [72], it is easy to create an Android application that broadcasts and detects beacons. However, recent changes in the iOS operating system will prevent iOS devices from doing that.

Chapter 6. Results

6.1 Machine Learning

The aim of this app's machine learning model was to accurately predict the distance between smartphones in order to determine a user's close contacts. For training data, the model used BLE RSSI readings from the MITRE Range-Angle structured dataset labelled with distances between smartphones. To test the effectiveness of our machine learning model, we firstly collected the Root Mean Squared Error (RMSE) gained from the training, test and cross-validation dataset for numerous feature sets. For each feature set, we collected the training, test, and validation RMSE gained from model type tested.

The MITRE Range-Angle structured dataset consists of a series of Bluetooth advertisements collected by smartphones according to the MITRE Structured Contact Tracing Protocol. The dataset was submitted as part of an effort to enhance contact tracing technology by the Private Automated Contact Tracing (PACT) project. PACT is a project whose mission is to " enhance contact tracing in pandemic response by designing exposure detection functions in personal digital communication devices that have maximal public health utility while preserving privacy" [59].

The dataset consisted of 69 sessions, each of which followed the MITRE Structured Contact Tracing Protocol. During each session, there were two testers: the beacon and the receiver. The beacon stays in a single position for the duration of the session and possesses a smartphone that sends BLE signals. The receiver uses a smartphone and the BlueProximity app to receive and record BLE advertisements at various distances and angles from the beacon. For each session, each tester chose a location for their smartphone to be held (choosing from shirt pocket, front pants pocket, back pants pocket, in purse/bag, or in hand) and a body orientation (sitting or standing). The session took place in one of the following types of environment: a small room, a medium-sized room, a large room, a hallway, or outdoors [47]. We preprocessed the MITRE dataset for training by extracting timestamps and RSSI data from the advertisements, separating them by windows of time, converting each window into a series of features, and splitting the sets of features into training, validation and test datasets. Figure 6.1 illustrates this process.

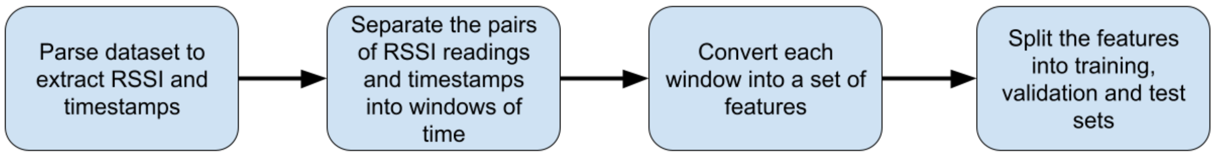


Figure 6.1: Machine Learning Preprocessing Procedure

Extracting the RSSI and timestamps was required because the dataset was in the form of unstructured files containing BLE advertisements. To obtain the RSSI data and the times of the advertisements in a usable format, we had to parse the files and extract the data from them. We separated the pairs of RSSI data and timestamps into windows of time to enable us to create machine learning models that used features based on windows of RSSI data rather than singular RSSI readings. This was done by iterating through the RSSI readings in order and adding them to the current window until the advertisements' timestamps indicated that a given amount of time had elapsed, and then creating a new window. To obtain the training data for the machine learning model, we computed the features of each window that would be used as input by the machine learning model. Finally, we split the dataset into training, validation, and test sets by randomly distributing them into a ratio of 60/20/20 respectively.

We firstly attempted to train the models using a single feature: the individual RSSI reading from each BLE signal. Of the feature sets tested, models that used this one had the highest error rates. This feature set will be further referred to as the Raw Set. Figure 6.2 shows the data collected for models using this feature set.

Regressor Type	**Lasso Linear Regression**	**Random Forests**
Best RMSE Hyperparameters	alpha = 0.0001	min_samples_leaf': 6
Best RMSE Gained	3.3018	3.2752
Test RMSE of best estimator	3.3194	3.2926
CV RMSE of best estimator	3.3282	3.2992
Test MAE of best estimator	2.7239	2.6585

CV MAE of best estimator	2.7262	2.6587
Best MAE Hyperparameters	alpha = 0.0001	min_samples_leaf': 4
Best MAE Gained	2.7011	2.6340
R^2 Score	0.2819	0.2945

Regressor Type	**LinearSVR (degree=2)**	**LinearSVR (degree=1)**
Best RMSE Hyperparameters	C': 3, 'loss': 'squared_epsilon_insensitive'	C': 1, 'loss': 'squared_epsilon_insensitive'
Best RMSE Gained	3.3021	3.3382
Test RMSE of best estimator	3.3194	3.3610
CV RMSE of best estimator	3.3282	3.3649
Test MAE of best estimator	2.7239	2.7330
CV MAE of best estimator	2.7261	2.7363
Best MAE Hyperparameters	C': 0.1, 'loss': 'epsilon_insensitive'	C': 10, 'loss': 'epsilon_insensitive'
Best MAE Gained	2.6423	2.6415
R^2 Score	0.2819	0.2659

Regressor Type	**K Nearest Neighbors Regressor**	**Radius Neighbors Regressor**
Best RMSE Hyperparameters	{'n_neighbors': 20, 'weights': 'distance'}	{'radius': 4, 'weights': 'distance'}
Best RMSE Gained	3.3484	3.2751
Test RMSE of best estimator	3.4099	3.2932
CV RMSE of best estimator	3.4063	3.2990
Test MAE of best estimator	2.7097	2.6595
CV MAE of best estimator	2.7059	2.6590
Best MAE Hyperparameters	{'n_neighbors': 20, 'weights': 'distance'}	{'radius': 8, 'weights': 'distance'}
Best MAE Gained	2.6802	2.6341
R^2 Score	0.2478	0.2944

Figure 6.2: Data collected using the raw set.

To lower error metrics, we then attempted to collect features by extracting them from a window of RSSI readings rather than from individual ones. These windows would contain signals received at intervals of a certain amount of seconds. For feature sets that used these windows, we tested them with window sizes of 0.5, 1.0, 1.5, and 2.0 seconds. The first feature set to use features extracted from these windows used the features in Figure 6.3.

Feature Name	**Feature Description**	**Formula**
Average	The average of all RSSI readings in the window.	$\frac{sum\ of\ all\ RSSI\ readings\ in\ the\ w}{number\ of\ RSSI\ readings\ in\ the\ w}$
Minimum	The minimum RSSI reading in the window.	$min(all\ RSSI\ readings\ in\ the\ window)$
Maximum	The maximum RSSI reading in the window.	$max(all\ RSSI\ readings\ in\ the\ window)$
Standard Deviation	The standard deviation of all RSSI readings in the window.	$\sqrt{\frac{\Sigma\ (x_i - average)^2}{number\ of\ RSSI\ readings\ in\ the\ window}}$, where x_i refers to each RSSI reading

Figure 6.3: A table displaying the features in Feature Set 1.

This feature set will be referred to as Feature Set 1. Models that used Feature Set 1 had lower validation RMSE values than models that used the Raw Set. After collecting error data from models that used Feature Set 1, we attempted to add more features to further decrease the models' error. This resulted in another feature set, which will be referred to as Feature Set 2. It used the features shown in Figure 6.4.

Feature Name	**Feature Description**	**Formula**
Average	The average of all RSSI readings in the window.	$\frac{sum\ of\ all\ RSSI\ readings\ in\ the}{number\ of\ RSSI\ readings\ in\ the}$

Minimum	The minimum RSSI reading in the window.	$min(all\ RSSI\ readings\ in\ the\ window)$
Maximum	The maximum RSSI reading in the window.	$max(all\ RSSI\ readings\ in\ the\ window)$
Standard Deviation	The standard deviation of all RSSI readings in the window.	$\sqrt{\frac{\Sigma\ (x_i - average)^2}{number\ of\ RSSI\ readings\ in\ the\ window}}$, where x_i refers to each RSSI reading
Median	The median RSSI reading in the window.	The (n+1)/2 th RSSI reading in the window when the RSSI readings are sorted.
Variance	The variance of all RSSI readings in the window.	$\frac{\Sigma\ (x_i - average)^2}{number\ of\ RSSI\ readings\ in\ the\ window}$, where x_i refers to each RSSI reading
Skewness	The skewness of all RSSI readings in the window.	$\frac{m_3}{{m_2}^{3/2}}$, where $m_i = \frac{1}{N}\sum_{n=1}^{N} (x_i - average)^i$, N = number of readings in the window, and x_i refers to each RSSI reading.
First Quartile	The first quartile RSSI reading.	The (n+1)/4 th RSSI reading in the window when the RSSI readings are sorted.
Third Quartile	The third quartile RSSI reading.	The 3(n+1)/4 th RSSI reading in the window when the RSSI readings are sorted.

Mean Absolute Difference	The mean absolute difference of all RSSI readings in the window.	$\frac{\Sigma \mid x_i - average \mid}{N}$, where x_i refers to each RSSI reading and N = number of RSSI readings in the window.
Kurtosis	The kurtosis of all RSSI readings in the window.	$\frac{c}{(standard\ deviation)^4}$, where $c = \frac{\Sigma\ (x_i - average)^4}{number\ of\ RSSI\ readings\ in\ the\ window}$
Range	Maximum - Minimum	Maximum - Minimum
Interquartile Range	Third Quartile - First Quartile	Third Quartile - First Quartile
Simple Moving Average	The simple moving average of the RSSI readings received.	$\frac{a_1+a_2+a_3+\ldots+a_n}{n}$, where a_1, a_2, …, a_n are the last n values of the average, and n is the number of values in the simple moving average window.

Figure 6.4: A table displaying the features in Feature Set 2.

Using Feature Set 2 also decreased models' validation RMSE. Figure 6.5 compares the best cross-validation(CV) RMSE obtained from using each combination of window size and feature set. K-fold cross-validation was used, with 10 folds.

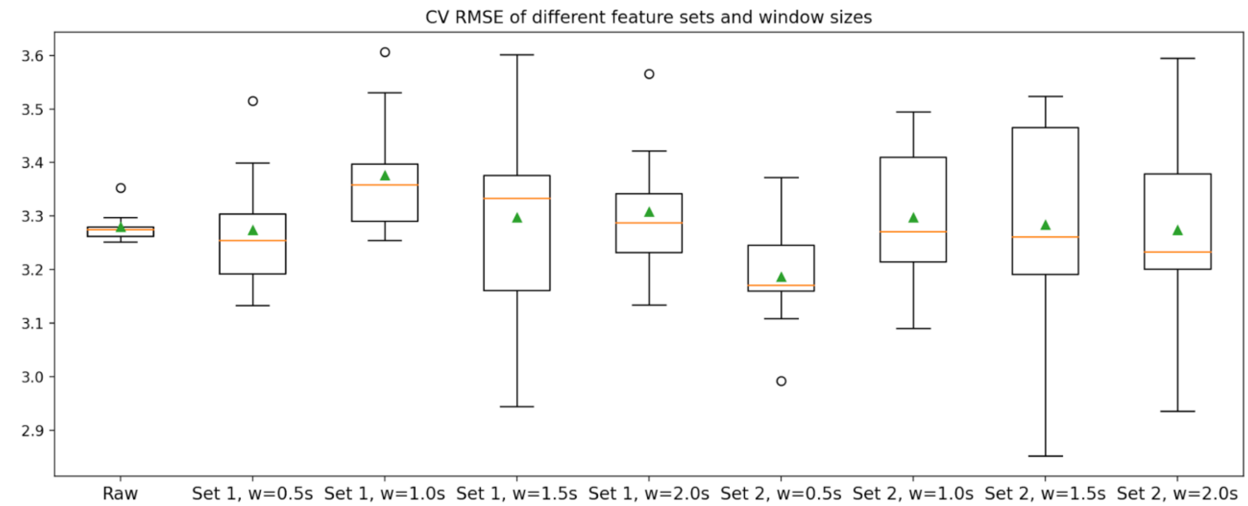


Figure 6.5: A chart illustrating the lowest CV RMSE gained from using each feature set and window size.

The chart shows that the lowest CV RMSE was obtained by using Feature Set 2 with a window size of 0.5 seconds. The model type that produced this RMSE was a random forests regressor. After identifying the feature set, window size, and model type that resulted in the lowest cross-validation RMSE, we attempted two methods to further optimize this model. The first of these was to add a simple moving average to the feature set. We did this to a random forests model trained using Feature Set 2 with a 0.5 second window size. We used that specific model because it produced the lowest CV RMSE and because random forests models tended to have the lowest validation RMSE values in other feature sets. To observe the impact of a simple moving average on CV RMSE, we collected CV RMSE when simple moving average windows ranging from 3 to 100 were used, and also when a cumulative moving average was used. Figure 6.6 displays how the simple moving average windows size affected CV RMSE.

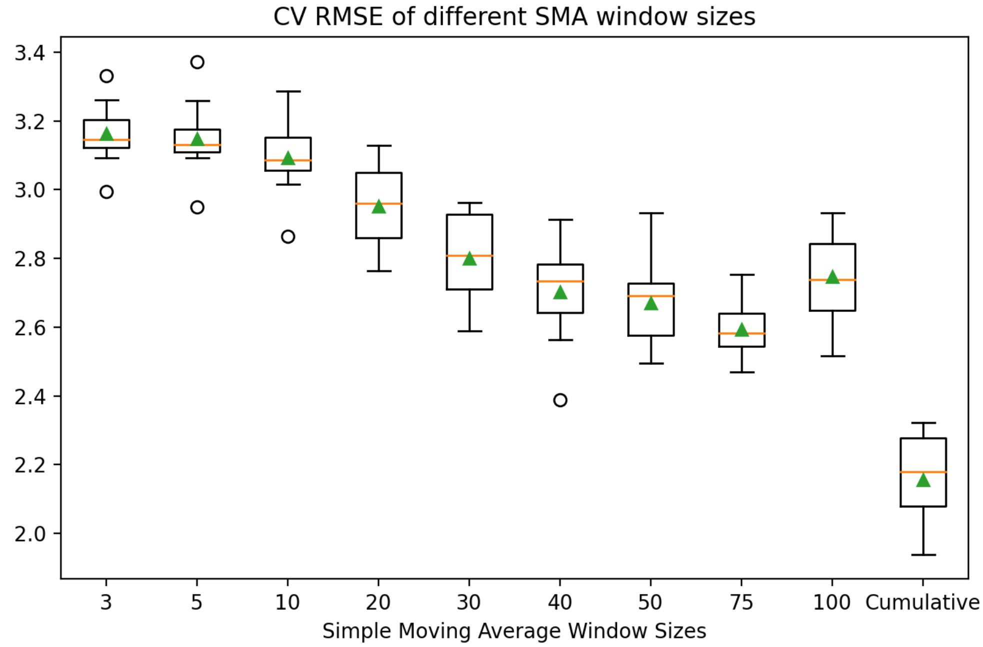


Figure 6.6: A chart displaying how simple moving average window size affected CV RMSE

Figure 6.6 shows that CV RMSE generally decreases as the simple moving average's window size increases. It also shows that the CV RMSE is at its lowest when a cumulative moving average is used. Therefore, using a simple moving average decreases the error rate of the model. The second optimization method was to train the data using overlapping windows of BLE signals, as the models had been previously trained using windows that did not overlap. We attempted to observe the impact of using overlapping windows during training by collecting RMSE when overlapping windows with amounts of overlap ranging from 10% to 50% were used. Overlapping windows is a method of creating windows such that the next window uses a percentage of the previous window. Figure 6.7 illustrates this concept. Figure 6.8 displays the result of training models with overlapping windows.

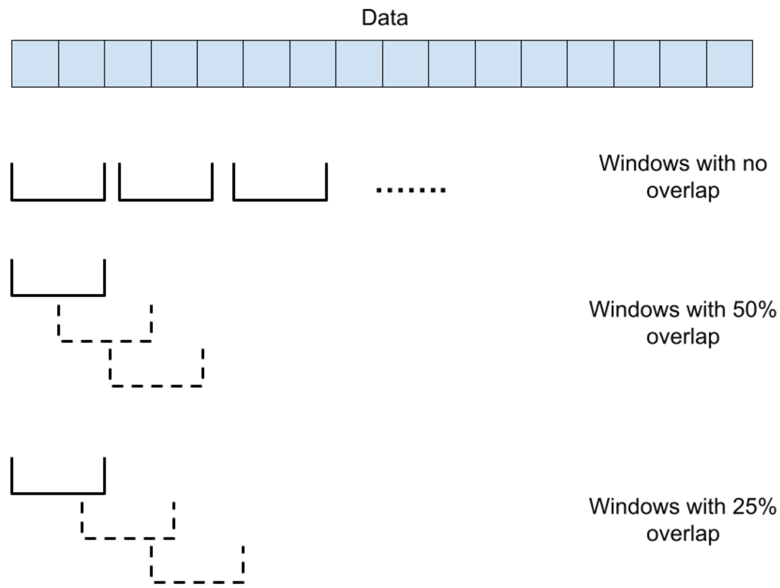


Figure 6.7: An illustration depicting overlapping windows.

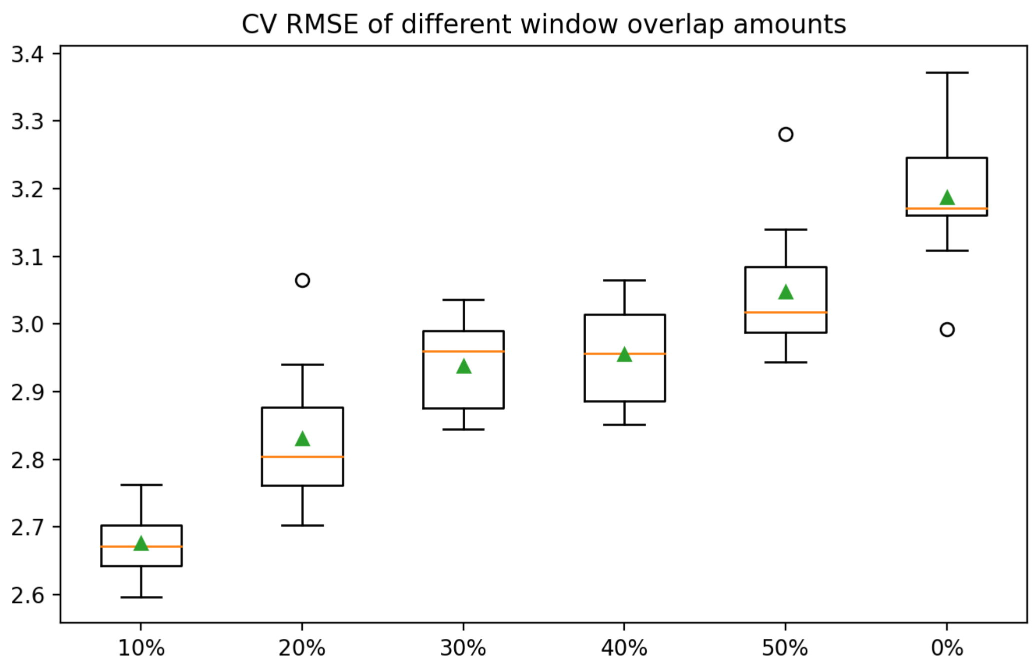


Figure 6.8: A chart showing how CV RMSE changes when the model is trained using different overlapping windows.

We finally attempted to improve the Feature Set 2 random forests model by modifying it to use both the simple moving average window and overlapping window amount that lead to the lowest CV RMSE readings. Figure 6.9 compares the best model that used Feature Set 2 to models that used simple moving average and overlapping windows.

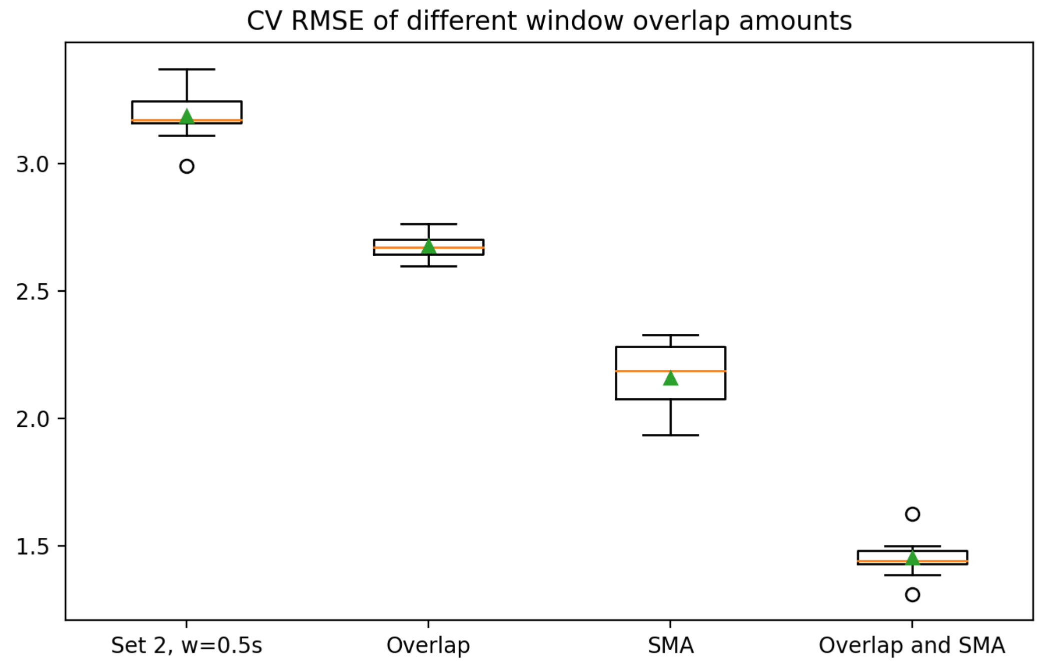


Figure 6.9: A chart showing how using simple moving average and overlapping windows impacts CV RMSE.

Using both simple moving average and overlapping windows resulted in the lowest CV RMSE. Before using these, the CV RMSE was 3.07433239 and the R squared score was -0.45458. Using simple moving average and overlapping windows decreased CV RMSE to 1.587660707, which was approximately a 50% decrease. The R squared score of the final model was -0.5933.

Figure 6.10 displays a graph of the actual distance versus the final model's predicted distances for data points in the test set. In addition, Figure 6.11 shows a chart of the final model's feature importances.

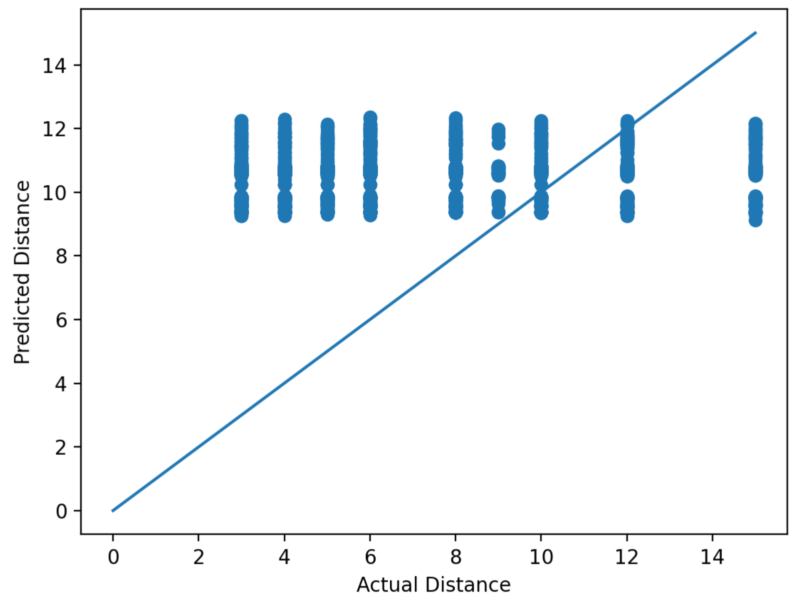


Figure 6.10: A scatter plot of actual distance against the final model's predicted test distances.

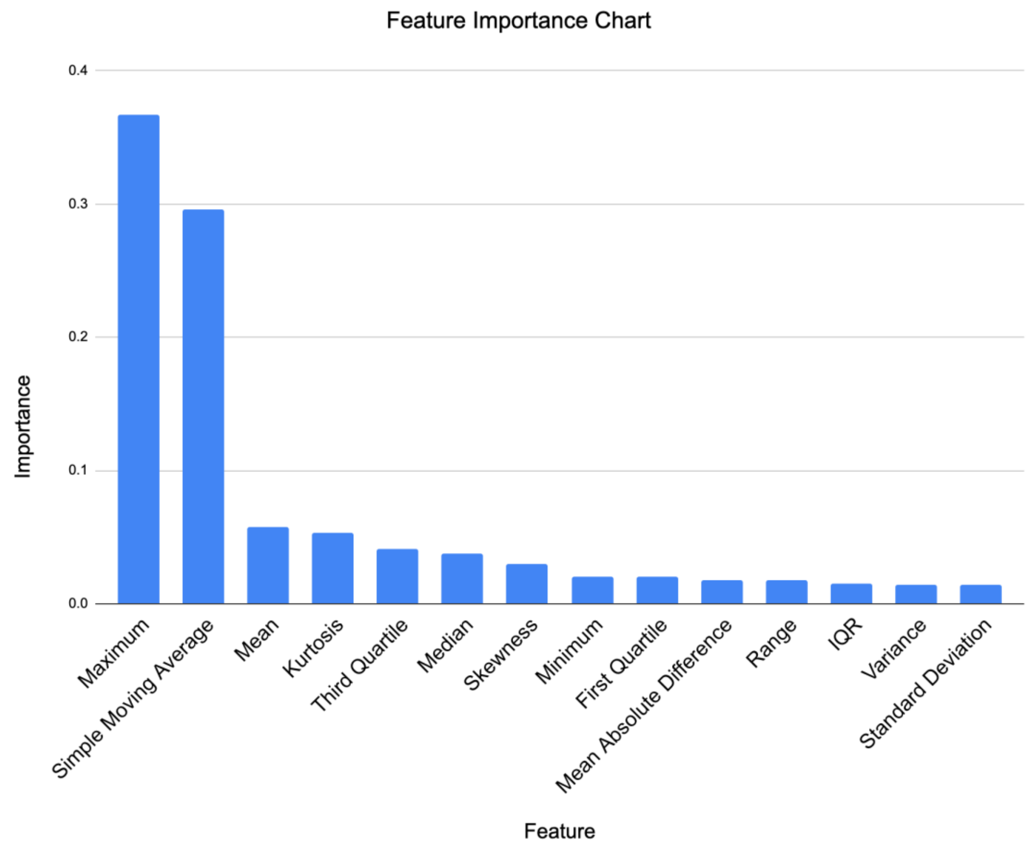


Figure 6.11: A chart displaying the feature importances of the final model.

6.2 Beacon Library Approach

We evaluated the beacons library functionality using two devices that were both broadcasting and scanning for beacons on a custom region. While experimenting, we kept some measurements of the actual distance of the two devices and the distance that was calculated by the library to find out whether it is giving accurate results. The results show an inconsistency between RSSI values and distance (in meters). As explained earlier, the beacon distance calculation is estimated, and the accuracy depends on the signal strength. The distance calculation is based on RSSI values and transmit power. Sometimes the calculation is not accurate and based on experimentation, setting the transmit power value to -59 dBm is optimal. Figure 6.12 shows how the distance, actual distance and RSSI are correlated.

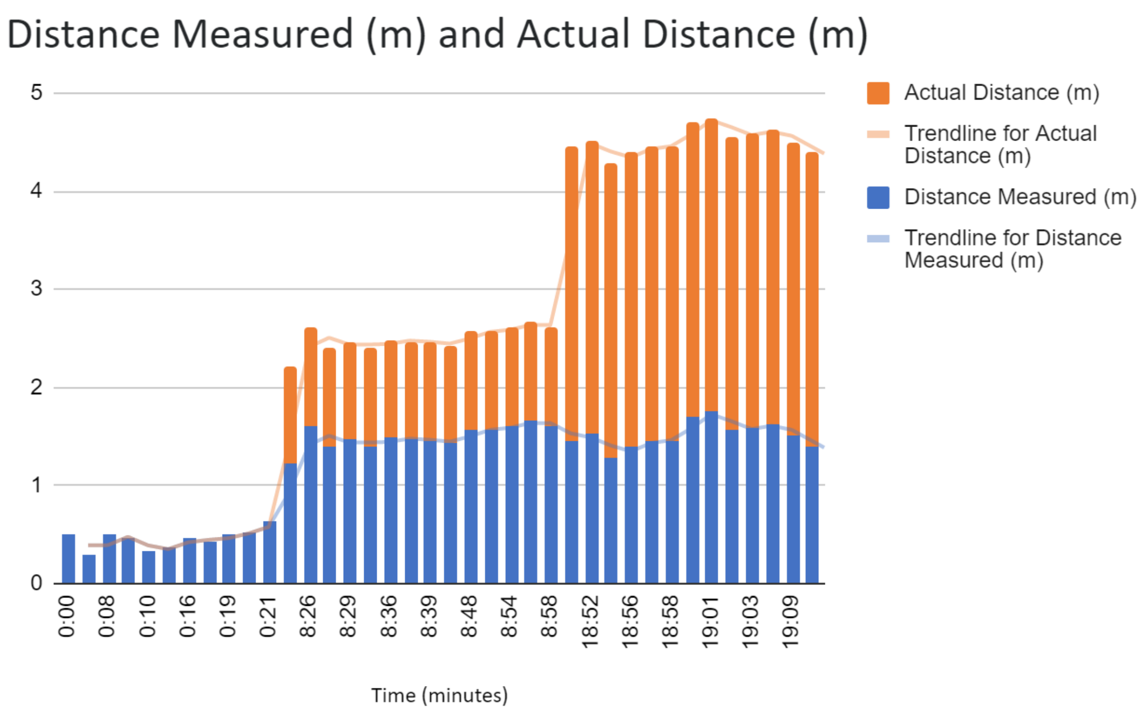


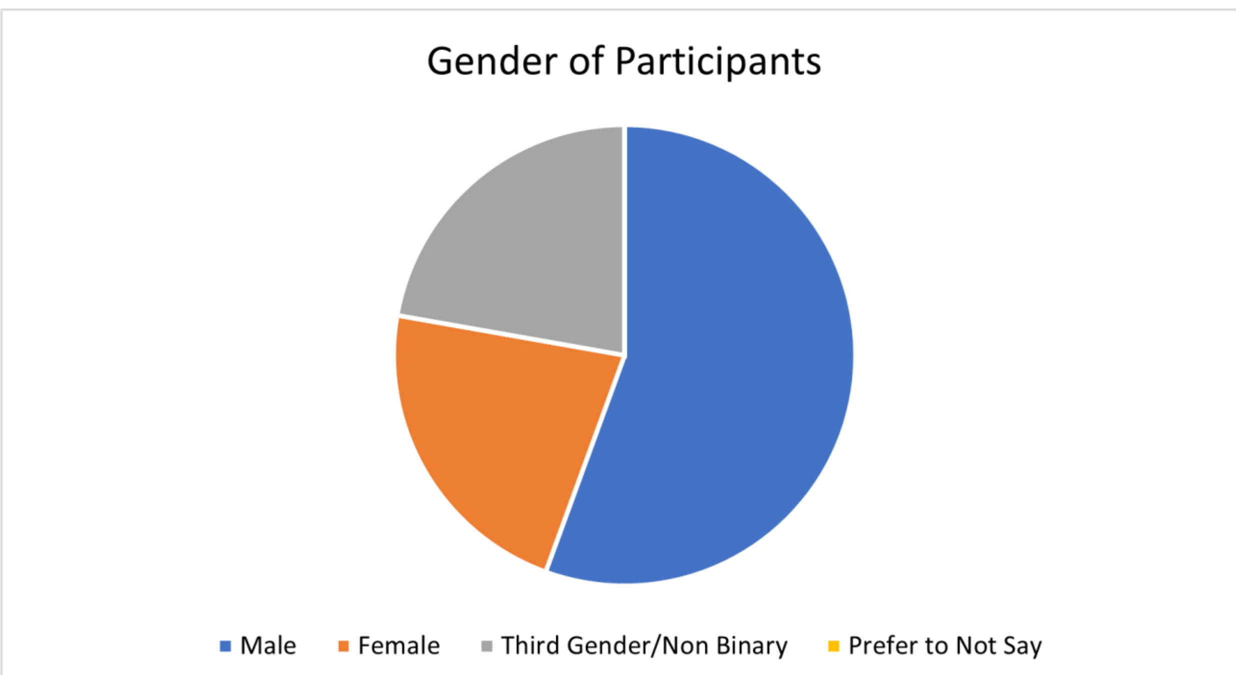
Figure 6.12: A diagram comparing distances from the AltBeacon library to actual distances.

This diagram shows that the distance that we are measuring is not close to the actual distance between the two phones. This indicates that the beacon library that we are using is not accurate when used with the specified parameters. The parameter "tx power" was set to -59 and we left it as is in order to perform the study.

6.3 App Evaluation Study

In order to gauge the effectiveness of our app's functionality and how accurately it estimated risk of contracting COVID19, we conducted a study so that the app would collect data in a real-world environment. In this study, 9 WPI students that were currently on campus were allowed to install the app on their Android smartphones. The study took place in two rounds in order to collect as much data as possible. In the first round, 5 students downloaded the app for a 5 day period from April 11, 2021 to April 16, 2021. In the second round, 4 students downloaded the app for 2 days from April 16, 2021. After downloading the app, the participants would answer the in-app questionnaire once. Once installed, the app would record the users' close contacts, in-app questionnaire responses, and risk scores for the duration of their study period. This data was then used to determine whether the app and the risk score calculation were working as intended.

We will now cover the results of this study. We firstly attempted to visualize the overall demographics of our study participants. Figure 6.13 shows the distribution of participants by gender and by class year. 5 of 9 (56%) of participants were Male. The remaining participants were evenly split between Female and Third Gender/Non Binary. In terms of class years, 6 of 9(67%) of all participants were Seniors. Freshman made up the next largest portion of participants, followed by Juniors. There were no sophomores in the study.



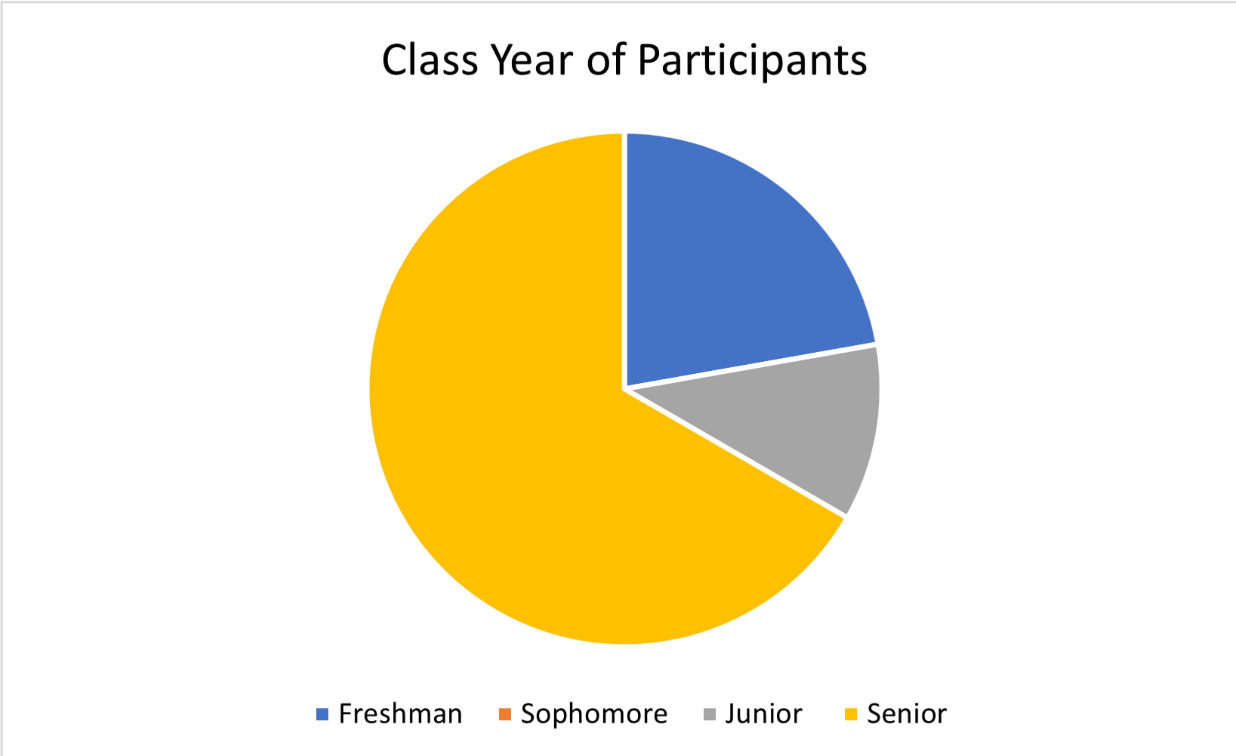


Figure 6.13: Charts showing how study participants were distributed by gender and class year.

We also analyzed all data collected from participants to gain a better understanding of it. We visualized participants' responses to specific questions from the in app questionnaire as shown below. First, we have the results for the question that asked users about their eating habits. We found that this question had the most variety in responses from participants although the most frequent answer was the participant 'Leaves the house to get groceries or take out 3-4x a month.' This variation was likely due to the different housing and food options for students from freshman dorms with full meal plans, upperclassmen dorms with kitchens and optional meal plans, and off campus apartments with full kitchens. Figure 6.14 shows the distribution for this questionnaire question.

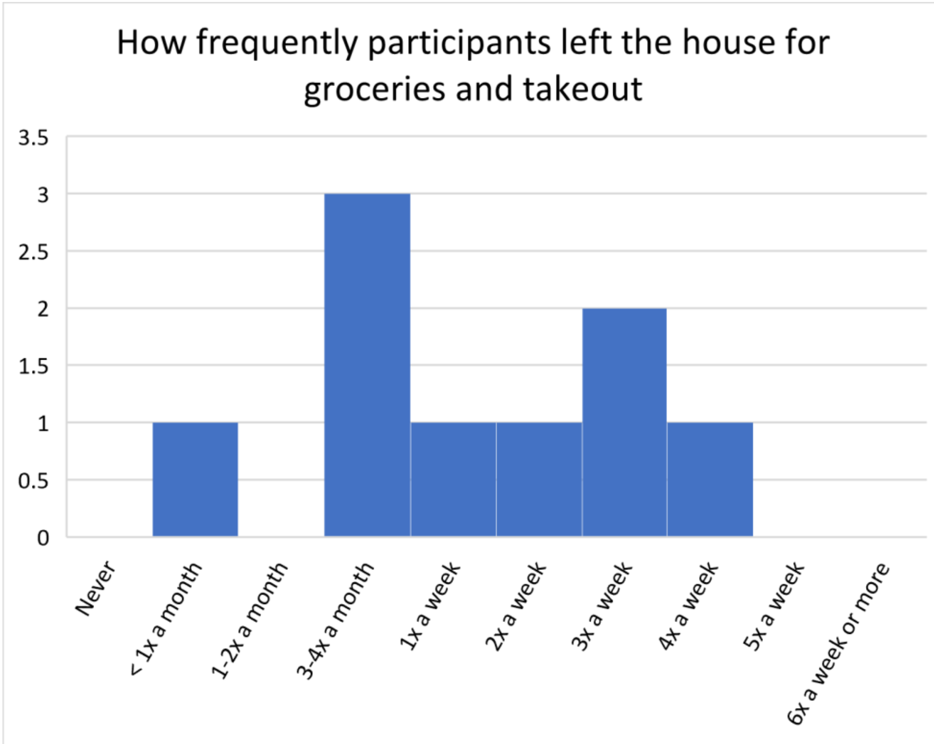


Figure 6.14: The frequency that participants left the house for groceries and takeout

Next, we have results about how frequently participants visited campus for course-related reasons. The results for this question are interesting because they have results on the highest end ‘6x a week or more’ and the lowest end ‘Never.’ This is likely due to the variety of course offerings this semester - some students had a completely virtual course load and others may have all classes in-person. The most common response that 5 out of 9 participants choose was ‘Going to campus 2-3x a week for course related reasons.’ Figure 6.15 displays the distribution of responses for going on campus.

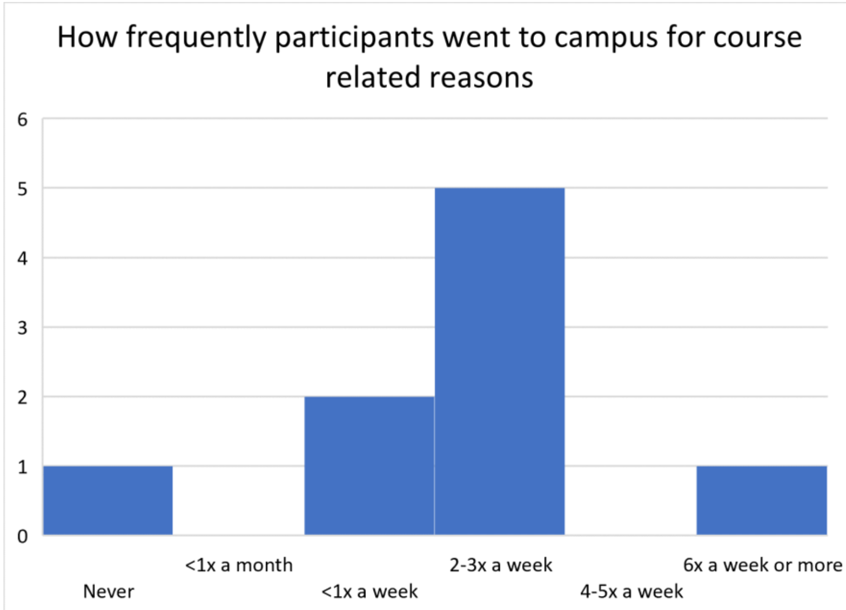


Figure 6.15: The frequency that participants when to campus for course related reasons

Finally, below is the histogram displaying the results for the question regarding participants' mask wearing habits. The responses to this question were very low among all participants with the largest number of participants choosing ‘Wears a mask 100% of the time.’ These results are similar for the Social Distancing question and the Transportation Use question. This indicates that participants had very safe behaviors in these three areas. Figure 6.16 shows the distribution of responses to the questionnaire question about mask wearing.

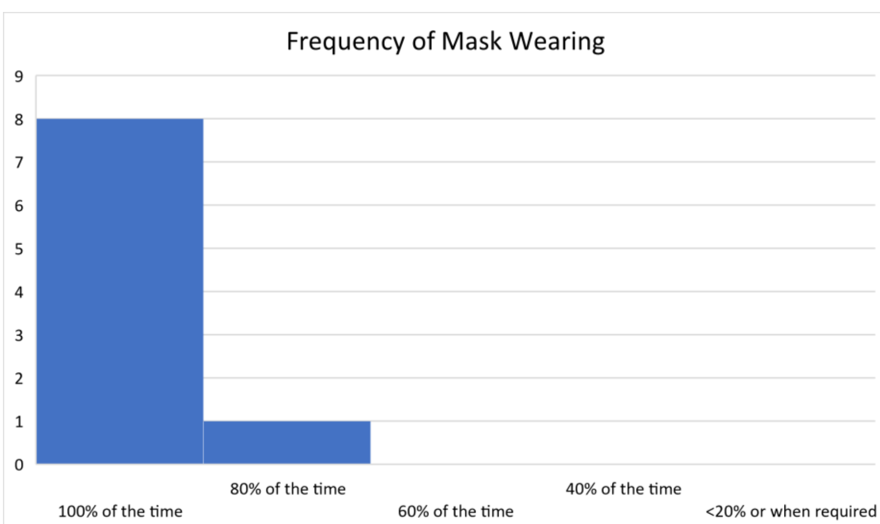


Figure 6.16: The frequency with which participants wore masks

Another crucial part of the data collection during the study was the number of contacts each participant had. Unfortunately, there were no close contacts recorded by the application during either one round in the study, therefore, there are no results.

To ensure that the risk scores given by the app accurately represented our participants' level of risk and were in line with our vision for the risk scores, we also visualized the distribution of risk scores. In general, there is a trend towards lower risk scores. They ranged between 14%-36% and the average risk score was 25.6% Figure 6.17 displays this.

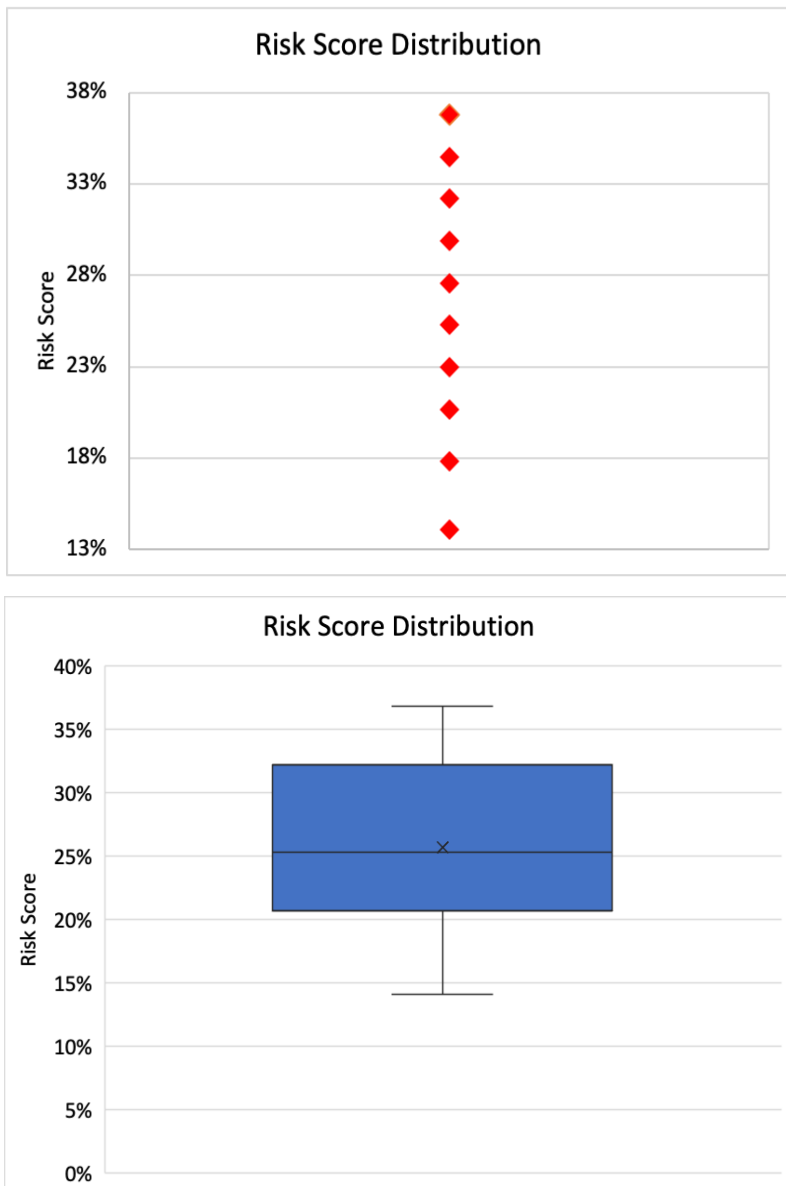


Figure 6.17: The distribution of risk scores among participants

In Figure 6.18, we can observe that the risk score of the male category has the largest range while the female category has the smallest one. We can conclude that the subjects belonging to the male and non-binary / third gender categories had higher risk scores.

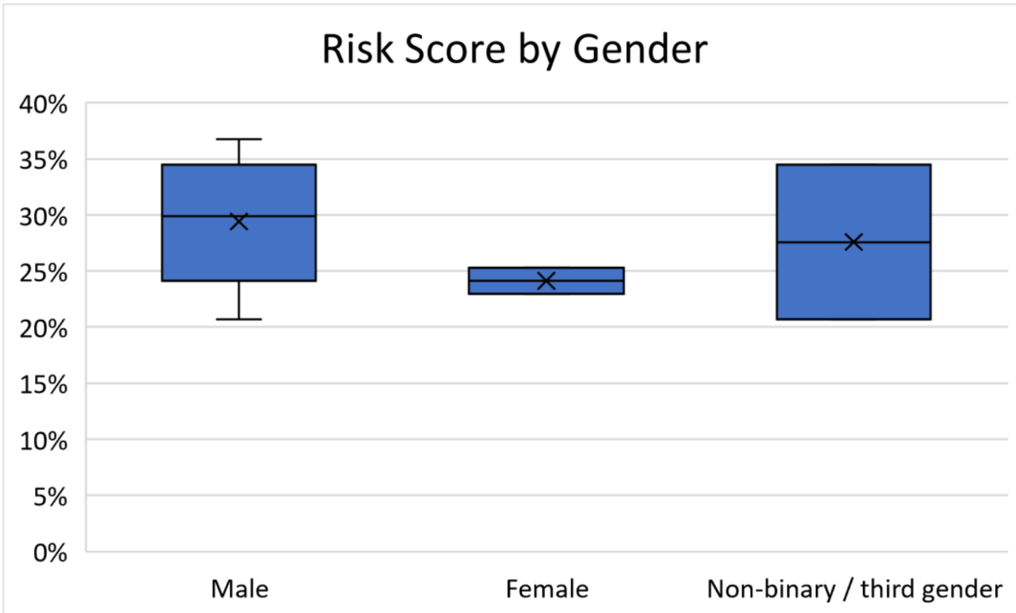


Figure 6.18: The distribution of risk scores by gender

Figure 6.19 shows the distribution of risk scores divided by class year. The most significant one being the one for seniors since two thirds of the subjects were seniors, as a result the graph for seniors is very similar to the graph for the whole study. The graph confirms most users had a risk score between 20%-30%.

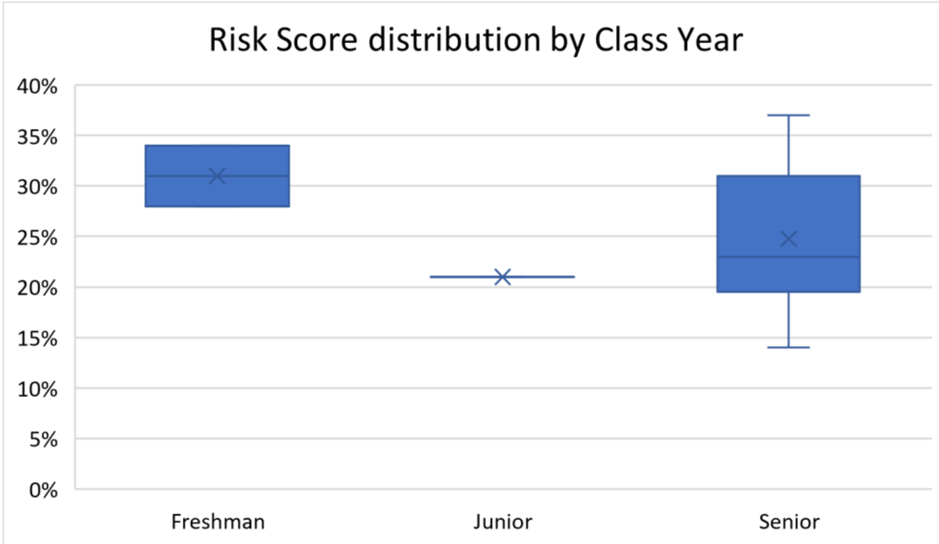


Figure 6.19: The distribution of risk scores by class year

In conclusion, we found that the sum of all questionnaire values most closely correlated with a user's calculated risk score. As one's questionnaire sum increases, the risk score tends to increase as well. This relationship applied to all groups of students that participated in the study. Figure 6.20 displays this relationship in a plot of risk score against the sum of questionnaire values. The data points that fall outside of this pattern have lower risk scores because they were calculated over a multi-day period and were subjected to the rolling average function of our risk score formula. A linear regression line was fitted to the data with an R^2 value of 0.6166. The Root Mean Squared Error (RSME) between the data points and the trendline is 0.0409.

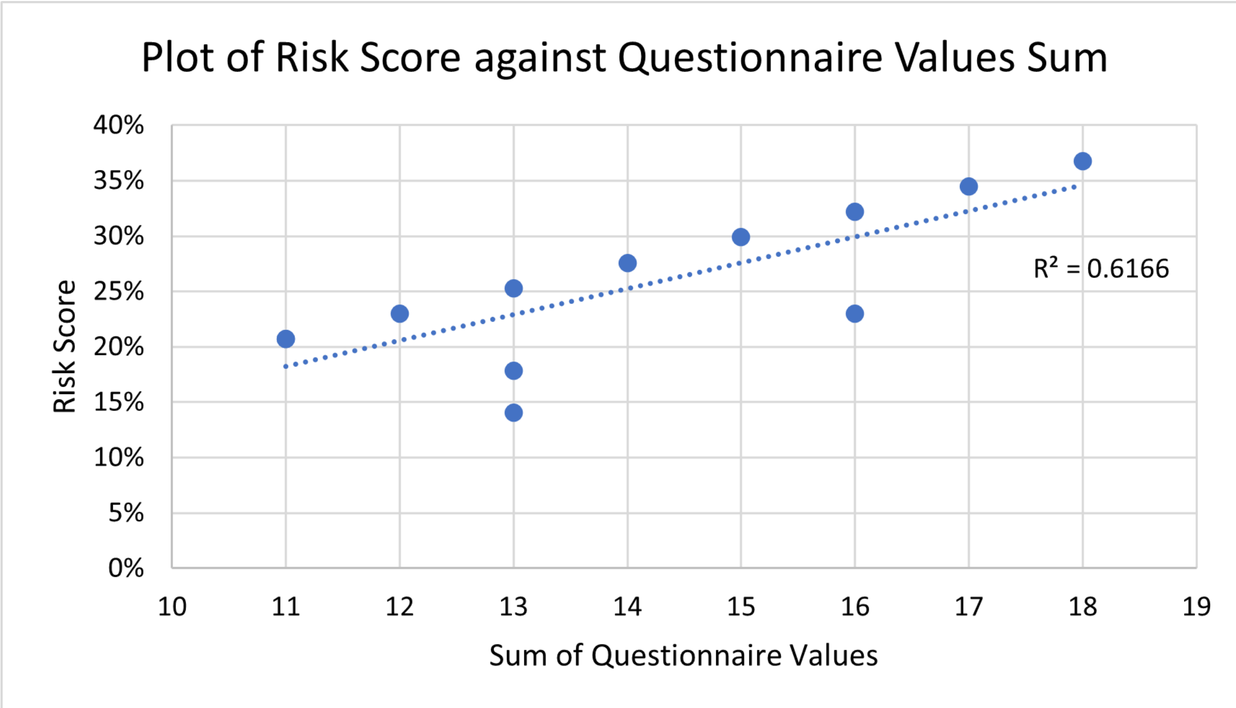


Figure 6.20: Scatter plot showing relationship between sum of questionnaire values and risk score

Chapter 7. Discussion

7.1 Successes

- All participants displayed a trend towards lower risk scores, with all risk scores ranging from 14% to 36%.
- There are two factors that could have caused this: the participants' questionnaire values and the number of contacts each participant had. As discussed in Chapter 6, most participants gave answers with low values for three of the five questionnaire questions, which would significantly decrease their total questionnaire sum. Therefore, the participants' low risk scores correlate with their questionnaire values.
- There are two factors that could have caused this: the participants' questionnaire values and the number of contacts each participant had. As discussed in Chapter 6, most participants gave answers with low values for three of the five questionnaire questions, which would significantly decrease their total questionnaire sum. Therefore, the participants' low risk scores correlate with their questionnaire values. As for contacts, there were none recorded during the study.

7.2 Challenges

- No contacts were recorded during the study, which prevents us from determining whether the risk scores correlate with participants' numbers of contacts. This could have been caused by participants not being in contact with each other or by a lack of BLE scanning by their smartphones.
- We observed during the study that specific phone models with Android 11 installed had battery optimization features that would aggressively close background apps. Numerous participants owned such phones, and the features prevented the app from running. This problem caused phones to not scan for close contacts at all and prevented them from recording risk scores.
- According to the plot of distances against the machine learning model's predicted distances, the model predicts long distances, even when the actual distance is short. This

may have been caused by an imbalance between longer distances and shorter distances in the dataset. The model's feature set or model type could have also led to this.

7.3 Limitations

- We were not able to find whether the risk score correlated with numbers of contacts because there were no contacts recorded during the study.
- There was a small number of participants during the study, which may have led to the trend of low questionnaire values.
- A close contact between two people will only be measured if both have the Goatvid app on their phones. Since both studies only had a small amount of subjects, the likelihood of two users being less than 6 feet away for more than 15 minutes was very low.
- The Android 11 problem prevented the smartphones from regularly recording risk scores and scanning for contacts.

7.4 Future Work

- As stated in Chapter 7, the battery optimization settings on Android 11 phones caused significant obstacles when testing the effectiveness of our app. Our next steps would be researching the Android 11 issue and finding a solution that allows the app to run continuously in the background. Next, we would test the app on a wide variety of phones from different brands and models as well as all Android versions. Finally, we would re-run the study with a much larger number of participants to ensure that close contacts could be detected.
- However, future Android versions may have even stronger battery optimization features which would cause the same issues to arise. An alternative solution could be to create a hardware device that can connect to the user's cellphone and the device can measure close contacts. The device would allow us to work around the battery optimization issue and allow users to have constant close contact detection. In turn this would also allow the risk score algorithm to work as intended and improve its accuracy. The next iteration of the project could then focus on improving the algorithm if necessary, with the complete results.

- In addition, we could display the server database contents within the mobile application. For example, using the current subjectID, a list of past risk scores could be displayed. We could also create a places table in the server and populate it with user data in order to have a better understanding of where close contacts occur. Additionally, we could further test the Beacon service that is running in the background in order to figure out what happens if there are more than one device in the area.

Chapter 8. Conclusion

Our work explores the idea of predicting one's risk of exposure to COVID-19 using smartphones. This establishes the feasibility of such a project as well as serves as a baseline for future projects in this area. The best achievement in the project was the development of the risk score formula which took into account a user's behaviors, contacts, location and past risk score to produce a singular score estimating their risk of exposure to COVID-19. In addition, we implemented passive close contact detection using the AltBeacon library. Using these beacons, we created a Machine Learning model that could estimate distances between two phones with a CV RMSE of 1.587660707. The final model's R squared score was -0.5933. During the study, the risk score seemed to adequately measure risk of COVID-19 exposure. However, due to the challenges facing the study, further testing is needed to confirm this.

Bibliography

[1] aaronlinsky. "What is Hansel?" https://github.com/gethansel/hansel (accessed May 2, 2021).

[2] Airship. "Getting Started." https://docs.airship.com/platform/android/getting-started/ (accessed 2021).

[3] E. Alpaydin and F. Bach, Introduction to Machine Learning. Cambridge, UNITED STATES: MIT Press, 2014.

[4] H. Alsdurf, "COVI White Paper - Version 1.1," 2020.

[5] A. Ng. "Machine Learning." https://www.coursera.org/learn/machine-learning (accessed 2020).

[6] A. Madan, M. Cebrian, D. Lazer, and A. Pentland, "Social sensing for epidemiological behavior change," presented at the Proceedings of the 12th ACM international conference on Ubiquitous computing, Copenhagen, Denmark, 2010. [Online]. Available: https://doi.org/10.1145/1864349.1864394.

[7] N. Bambrick. "Support Vector Machines: A Simple Explanation." https://www.kdnuggets.com/2016/07/support-vector-machines-simple-explanation.html (accessed 2021).

[8] B. Ray. "Bluetooth Vs. Bluetooth Low Energy (BLE): What's The Difference?" @LinkLabsInc. https://www.link-labs.com/blog/bluetooth-vs-bluetooth-low-energy (accessed October 15, 2020).

[9] C19X. "C19X." https://github.com/c19x?tab=overview&from=2021-04-01&to=2021-04-30 (accessed May 2, 2021).

[10] CDC, "Back-To-College Tips Protect yourself from COVID-19," College-Poster-COVID-Tips-for-Students, Ed., ed. https://www.cdc.gov/coronavirus/2019-ncov/downloads/College-Poster-COVID-Tips-for-Students.pdf: CDC, 2020.

[11] CDC. "Operational Considerations for Adapting a Contact Tracing Program to Respond to the COVID-19 Pandemic." CDC. https://www.cdc.gov/coronavirus/2019-ncov/global-covid-19/operational-considerations-contact-tracing.html (accessed October 11, 2020).

[12] CDC. "What you should know about COVID-19 to protect yourself and others." https://www.cdc.gov/coronavirus/2019-ncov/downloads/2019-ncov-factsheet.pdf (accessed 2021).

[13] CDC. "COVID Data Tracker." https://covid.cdc.gov/covid-data-tracker/#cases_totalcases (accessed 2021).

[14] CDC. "Symptoms of Coronavirus." https://www.cdc.gov/coronavirus/2019-ncov/symptoms-testing/symptoms.html (accessed 2021).

[15] N. Chauhan. "Decision Tree Algorithm, Explained." https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html (accessed 2021).

[16] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.

[17] Corona-Warn-App. "Open-Source Project Corona-Warn-App." https://www.coronawarn.app/en/ (accessed October 15, 2020).

[18] Covi-ID. "Covi-ID." https://coviid.me/ (accessed 2020).

[19] L. COVID-19 RECoVERY CONSULTING. "COVID-19 Activity Risk Index." https://static1.squarespace.com/static/5ed1588a45f60d321b51953b/t/5f1de89a86ffed423f15d3fe/1595795612665/covid19reopen.com%3Acovid-19-risk-index.pdf (accessed 2021).

[20] Crowdless. "Crowdless." https://crowdlessapp.co/ (accessed October 15, 2020).

[21] Crowdless. "Crowdless." https://crowdlessapp.co/ (accessed October 15, 2020).

[22] O. DeMasi, S. Feygin, A. Dembo, A. Aguilera, and B. Recht, "Well-Being Tracking via Smartphone-Measured Activity and Sleep: Cohort Study," (in English), *JMIR Mhealth Uhealth*, Original Paper vol. 5, no. 10, p. e137, 2017, doi: 10.2196/mhealth.7820.

[23] D. Telekom. "Corona-Warn-App Open Source Project." https://www.coronawarn.app/en/ (accessed 2020).

[24] E. Gurley. "COVID-19 Contact Tracing." https://www.coursera.org/learn/covid-19-contact-tracing? (accessed 2020).

[25] ENISA, Privacy and data protection in mobile applications. ENISA, 2017.

[26] G. Ionescu, C. M. d. l. Osa, and M. Deriaz, "Improving distance estimation in object localisation with bluetooth low energy," *SENSORCOMM 2014 - 8th International Conference on Sensor Technologies and Applications*, 2014.

[27] Google. "Exposure Notifications: Using technology to help public health authorities fight COVID-19." Google. https://www.google.com/covid19/exposurenotifications/ (accessed 2020).

[28] H. Partners. "Mind the Gap." https://mindthegap.today/ (accessed 2020).

[29] G. M. Harari, N. D. Lane, R. Wang, B. S. Crosier, A. T. Campbell, and S. D. Gosling, "Using Smartphones to Collect Behavioral Data in Psychological Science: Opportunities, Practical Considerations, and Challenges," (in eng), *Perspectives on psychological science : a journal of the Association for Psychological Science*, vol. 11, no. 6, pp. 838-854, 2016, doi: 10.1177/1745691616650285.

[30] G. M. Harari, S. R. Müller, M. S. H. Aung, and P. J. Rentfrow, "Smartphone sensing methods for studying behavior in everyday life," *Current Opinion in Behavioral Sciences*, vol. 18, pp. 83-90, 2017/12/01/ 2017, doi: https://doi.org/10.1016/j.cobeha.2017.07.018.

[31] G. Iacurci. "Job losses remain ‘enormous’: Coronavirus unemployment claims are worst in history." https://www.cnbc.com/2020/07/09/coronavirus-unemployment-benefits-claims-are-the-worst-in-history.html (accessed 2020).

[32] I. T. R. Center. "Annual number of data breaches and exposed records in the United States from 2005 to 2020 (in millions)." https://www.statista.com/statistics/273550/data-breaches-recorded-in-the-united-states-by-number-of-breaches-and-records-exposed/ (accessed April 21, 2021).

[33] I. s. M. o. Health. "MohGovIL/hamagen-react-native: Israel ' s Ministry of Health ' s COVID-19 Exposure Prevention App." (accessed October 15, 2020).

[34] J. H. U. MEDICINE and C. R. CENTER. "COVID-19 United States cases by county." https://coronavirus.jhu.edu/us-map (accessed Feb. 21, 2021).

[35] M. Krangle. "MITRE Range-Angle (Structured)." https://github.com/mkrangle/MITRE-Range-Angle-Structured (accessed 2020).

[36] Laravel. "Database: Query Builder." https://laravel.com/docs/8.x/queries (accessed Mar. 1, 2021).

[37] Laravel. "Eloquent: Getting Started." https://laravel.com/docs/8.x/eloquent (accessed Mar. 1, 2021).

[38] D. J. Leith and S. Farell, "Coronavirus Contact Tracing: Evaluating The Potential Of Using Bluetooth Received Signal Strength For Proximity Detection," School of Computer Science & Statistics, Trinity College, Dublin, Ireland, 2020.

[39] Z. Lin, N. Shroff, H. Wen, D. Xuan, and Q. Zhao, "On the Accuracy of Measured Proximity of Bluetooth-based Contact Tracing Apps," Department of Computer Science and Engineering, The Ohio State University, 2020.

[40] C. Linder. "Can Bluetooth Contact Tracing Apps Really Help Fight Pandemics?" @PopMech.

https://www.popularmechanics.com/technology/security/a32108923/bluetooth-contact-tracing-app/ (accessed October 15, 2020).

[41] C. Linder. "Can Bluetooth Contact Tracing Apps Really Help Fight Pandemics?" (accessed September, 2020).

[42] M. Zastrow. "Coronavirus contact-tracing apps: can they slow the spread of COVID-19?" Nature Publishing Group. https://www.nature.com/articles/d41586-020-01514-2 (accessed October 15, 2020).

[43] Mathematica. "19 and Me: COVID-19 Risk Score Calculator." https://19andme.covid19.mathematica.org/ (accessed 2020).

[44] N. M. Michalak, O. Sng, I. M. Wang, and J. Ackerman, "Sounds of sickness: can people identify infectious disease using sounds of coughs and sneezes?," *Proceedings of the Royal Society B*, 2020.

[45] Milvum. "PrivateTracer." https://www.privatetracer.org/ (accessed May 3, 2021).

[46] M. T. Gap. "Mind The Gap." https://mindthegap.today/about (accessed October 15, 2020).

[47] MIT. "Structured Contact Tracing Protocol, V. 2.0." https://mitll.github.io/PACT/files/Structured%20Contact%20Tracing%20Protocol,%20V.%202.0%20(1.5).pdf (accessed Mar 13, 2021).

[48] MohGovIL. "Hamagen." https://github.com/MohGovIL/hamagen-react-native (accessed 2020).

[49] P. Nair. "The dummy’s guide to MFCC." Medium. https://medium.com/prathena/the-dummys-guide-to-mfcc-aceab2450fd (accessed 2020).

[50] K. A. Nguyen and Z. L. Luo, "Cover Your Cough: Detection of Respiratory Events with

Confidence Using a Smartwatch," Computer Science Department, University of London, Surrey, United Kingdom, 2018.

[51] PEPP-PT. "HOME | PEPP-PT." https://www.pepp-pt.org (accessed October 15, 2020).

[52] PEPP-PT. "Pan-European Privacy-Preserving Proximity Tracing." https://www.pepp-pt.org/ (accessed 2020).

[53] P. R. Center. "Mobile Fact Sheet." https://www.pewresearch.org/internet/fact-sheet/mobile/ (accessed 2021).

[54] R. Networks. "Android Beacon Library." https://altbeacon.github.io/android-beacon-library/ (accessed Mar. 1, 2021).

[55] R. Networks. "Distance estimates." https://altbeacon.github.io/android-beacon-library/distance-calculations.html (accessed Mar. 1, 2021).

[56] R. Networks. "Using a Foreground Service." https://altbeacon.github.io/android-beacon-library/foreground-service.html (accessed Mar. 1, 2021).

[57] B. Rashidi, "Smartphone User Privacy Preserving through Crowdsourcing," Department of Computer Science, Virginia Commonwealth University, Richmond, Virginia, 2018.

[58] B. Ray. "Bluetooth Vs. Bluetooth Low Energy: What's The Difference?" (accessed November 01, 2020).

[59] R. Rivest, L. Ivers, I. Soibelman, and M. Zissman. "PACT Mission and Approach." https://pact.mit.edu/wp-content/uploads/2020/05/PACT-Mission-and-Approach-2020-05-19-.pdf (accessed Mar 13, 2021).

[60] S. Bertuletti, A. Cereatti, M. Caldara, M. Galizzi, and U. D. Croce, "Indoor distance estimated from Bluetooth Low Energy signal strength: Comparison of regression models," presented at the 2016 IEEE Sensors Applications Symposium (SAS), Catania, Italy, 2016. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7479899/authors#authors.

[61] M. Scudellari. "COVID-19 Digital Contact Tracing: Apple and Google Work Together as MIT Tests Validity." @IEEESpectrum. https://spectrum.ieee.org/the-human-

os/biomedical/devices/covid19-digital-contact-tracing-apple-google-mit-tests-validity (accessed October 15, 2020).

[62] Y. Shi, H. Lie, Y. Wang, M. Cai, and W. Xu, "Theory and Application of Audio-Based Assessment of Cough," *Journal of Sensors*, vol. 2018, 2017.

[63] T. Srivastava. "K Nearest Neighbor." https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/ (accessed 2021).

[64] T. T. Challenge. "NIST Pilot TC4TL Challenge." https://tc4tlchallenge.nist.gov/ (accessed 2021).

[65] U. S. B. o. L. Statistics. "Civilian Unemployment Rate." https://www.bls.gov/charts/employment-situation/civilian-unemployment-rate.htm (accessed 2021).

[66] u/ollien. "WPI COVID Dashboard Tracking." https://docs.google.com/spreadsheets/d/1nlu7V6P-D73tY5jGb0WGDU2XLUMqEV8pvwpm_2bGzVc/edit#gid=0 (accessed 2020).

[67] WeHealth. "Covid Watch Arizona." https://www.wehealth.org/arizona (accessed May 2, 2021).

[68] A. M. Wilson et al., "Quantifying SARS-CoV-2 infection risk within the Google/Apple exposure notification framework to inform quarantine recommendations," *medRxiv*, p. 2020.07.17.20156539, 2020, doi: 10.1101/2020.07.17.20156539.

[69] W. H. Organization, "Digital tools for COVID-19 contact tracing," Annex: Contact tracing in the context of COVID-19, 2020.

[70] WPI. "COVID Testing Dashboard." https://www.wpi.edu/we-are-wpi/health-and-safety/dashboard (accessed February 19, 2021).

[71] Y. Shi, H. Liu, Y. Wang, M. Cai, and W. Xu, "Theory and Application of Audio-Based Assessment of Cough," *Hindawi*, 2018.

[72] D. Young. "Hacking With Contact Tracing Beacons." http://www.davidgyoungtech.com/2020/04/24/hacking-with-contact-tracing-beacons (accessed Mar. 9, 2021).

[73] M. Zastrow. "Coronavirus contact-tracing apps: can they slow the spread of COVID-19?" https://www.nature.com/articles/d41586-020-01514-2 (accessed 2020).

[74] M. Zens, A. Brammertz, J. Herpich, N. Südkamp, and M. Hinterseer, "App-Based Tracking of Self-Reported COVID-19 Symptoms: Analysis of Questionnaire Data," (in English), *J Med Internet Res,* Short Paper vol. 22, no. 9, p. e21956, 2020, doi: 10.2196/21956.

Appendices:

A. In-App Questionnaire

1. Which of the options below align most closely with the way in which you attend classes? (Course related reasons include lectures, labs, office hours, group meetings, project work, etc.)

    a. Never leave the house for course related reasons

    b. Going to campus < 1x a week for course related reasons

    c. Going to campus 1x a week for course related reasons

    d. Going to campus 2-3x a week for course related reasons

    e. Going to campus 4-5x a week for course related reasons

    f. Going to campus 6x a week or more for course related reasons

2. Which of the options below align most closely with the way in which you get food?

    a. Never leave the house to pick up groceries or take out out
    b. Leaves the house to get groceries or take out < 1x a month
    c. Leaves the house to get groceries or take out 1-2x a month
    d. Leaves the house to get groceries or take out 3-4x a month
    e. Leaves the house to get groceries or take out 1x a week
    f. Leaves the house to get groceries or take out 2x a week
    g. Leaves the house to get groceries or take out 3x a week
    h. Leaves the house to get groceries or take out 4x a week
    i. Leaves the house to get groceries or take out 5x a week
    j. Leaves the house to get groceries or take out 6x a week or more

3. Which of the options below align most closely with the frequency that you attend campus for non-course work related reasons? (Non course related reasons include work, food, seeing friends, etc.)

    a. Only going to campus for COVID testing

    b. Going to campus 1x a week (other than for testing)

c. Going to campus 2-3x a week (other than for testing)

d. Going to campus 4-5x a week (other than for testing)

e. Going to campus 6x a week or more (other than for testing)

4. Which of the options below most closely aligns with your mask wearing behavior?

a. Wears a mask 100% of the time

b. Wears a mask 80% of the time

c. Wears a mask 60% of the time

d. Wears a mask 40% of the time

e. Only wearing a mask when required or < 20% of the time

5. Which of the following options most closely relates to your social distancing behavior?

a. Practices social distancing 100% of the time

b. Practices social distancing 80% of the time

c. Practices social distancing 60% of the time

d. Practices social distancing 40% of the time

e. Only practices social distancing when required or < 20% of the time

6. Which of the options below align most closely with how you use public transportation? This includes buses, trains, planes, rideshares (Uber), taxis or any method other than a personal vehicle.

a. Never using public transportation

b. Using public transportation < 1x a month

c. Using public transportation 2-3x a month

d. Using public transportation < 1x a week

e. Using public transportation 1x a week

f. Using public transportation 2-3x a week

g. Using public transportation 4-5x a week

h. Using public transportation 6x a week or more

B. Behavior Risk Score Values

The table below presents the associated value of each multiple choice of each question from the questionnaire.

<table>
<tr><th colspan="11">Questionnaire Value relations</th></tr>
<tr><td>Question 1: Class attending value</td><td>Never leave the house for course related reasons</td><td>Going to campus < 1x a week for course related reasons</td><td colspan="2">Going to campus 1x a week for course related reasons</td><td colspan="2">Going to campus 2-3x a week for course related reasons</td><td colspan="2">Going to campus 4-5x a week for course related reasons</td><td colspan="2">Going to campus 6x a week or more for course related reasons</td></tr>
<tr><td>Value</td><td>0</td><td>1</td><td colspan="2">2</td><td colspan="2">3</td><td colspan="2">4</td><td colspan="2">5</td></tr>
<tr><td>Question 2: Food source value</td><td>Never leave the house to pick up groceries or take out out</td><td>Leaves the house to get groceries or take out < 1x a month</td><td>Leaves the house to get groceries or take out 1-2x a month</td><td>Leaves the house to get groceries or take out 3-4x a month</td><td>Leaves the house to get groceries or take out 1x a week</td><td>Leaves the house to get groceries or take out 2x a week</td><td>Leaves the house to get groceries or take out 3x a week</td><td>Leaves the house to get groceries or take out 4x a week</td><td>Leaves the house to get groceries or take out 5x a week</td><td>Leaves the house to get groceries or take out 6x a week or more</td></tr>
<tr><td>Value</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr>
<tr><td>Question 3: Campus visit value</td><td colspan="2">Only going to campus for COVID testing</td><td colspan="2">Going to campus 1x a week (other than for testing)</td><td colspan="2">Going to campus 2-3x a week (other than for testing)</td><td colspan="2">Going to campus 4-5x a week (other than for testing)</td><td colspan="2">Going to campus 6x a week or more (other than for testing)</td></tr>
<tr><td>Value</td><td colspan="2">1</td><td colspan="2">2</td><td colspan="2">3</td><td colspan="2">4</td><td colspan="2">5</td></tr>
<tr><td>Question 4: Mask value</td><td colspan="2">Wears a mask 100% of the time</td><td colspan="2">Wears a mask 80% of the time</td><td colspan="2">Wears a mask 60% of the time</td><td colspan="2">Wears a mask 40% of the time</td><td colspan="2">Only wearing a mask when required or < 20% of the time</td></tr>
<tr><td>Value</td><td colspan="2">1</td><td colspan="2">2</td><td colspan="2">3</td><td colspan="2">4</td><td colspan="2">5</td></tr>
<tr><td>Question 5: Social distance value</td><td colspan="2">Practices social distancing 100% of the time</td><td colspan="2">Practices social distancing 80% of the time</td><td colspan="2">Practices social distancing 60% of the time</td><td colspan="2">Practices social distancing 40% of the time</td><td colspan="2">Only practices social distancing when required or < 20% of the time</td></tr>
<tr><td>Value</td><td colspan="2">1</td><td colspan="2">2</td><td colspan="2">3</td><td colspan="2">4</td><td colspan="2">5</td></tr>
<tr><td>Question 6: Transportation value</td><td>Never using public transportation</td><td>Using public transportation < 1x a month</td><td>Using public transportation 2-3x a month</td><td>Using public transportation < 1x a week</td><td>Using public transportation 1x a week</td><td>Using public transportation 2-3x a week</td><td colspan="2">Using public transportation 4-5x a week</td><td colspan="2">Using public transportation 6x a week or more</td></tr>
<tr><td>Value</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td colspan="2">6</td><td colspan="2">7</td></tr>
</table>

The table below illustrates where the Risk score gets the raw data from

Raw Data to be Used As Partial Risk Score	**Source**
Number of close contacts (15 minutes within 6ft)	Beacons/ML model
Time spent in close contact (number of minutes)	Beacons/ML model
Questionnaire input	Input from User

C. Place Type Risk Score Values

The location categories below are describing what each location label represents. Those labels are pre-defined by Google's API. We are receiving those from the Places API.

Location	Place Type API Labels	Risk Scores
Bar	bar	9
Restaurant	restaurant	9
Night Club	night_club	9
Movie Theater	movie_theater	8
Supermarket	supermarket	5
Shopping Mall	shopping_mall	5
Natural Feature	natural_feature	5

D. Contact Tracing App Research (link here)

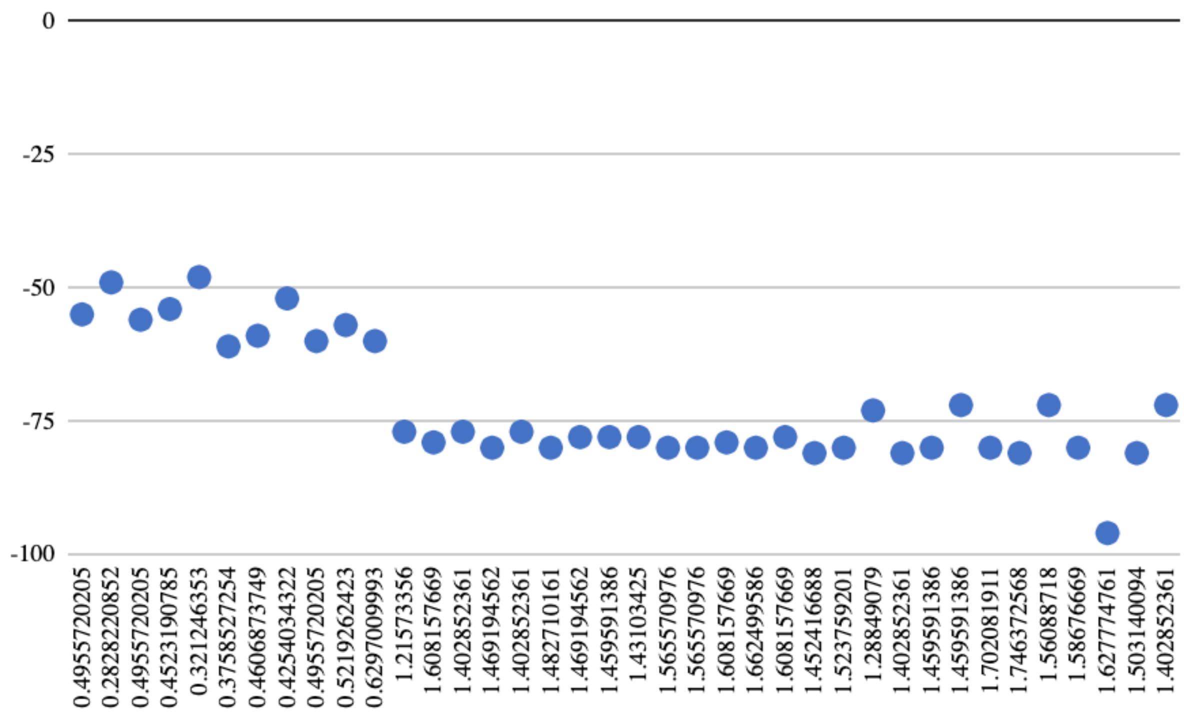
As part of the literature review for this project, we researched numerous contact tracing apps to find potential features for our app, and ways of implementing those features. The apps that we found during our research and documented are displayed in the following tables.

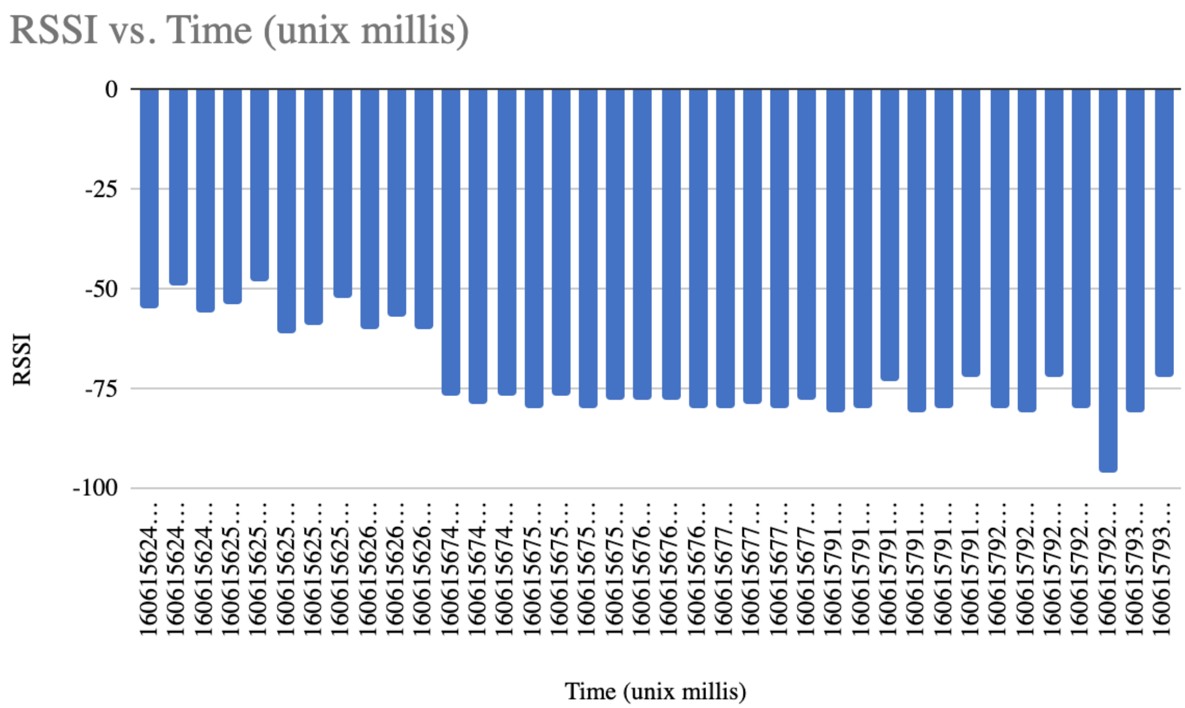
App Name (hyperlink to github or website)	App Category	Platform (include web)	Who Made It	Completed?	Open Source?	Purpose
my app						
Covid World	Database	Android	Not Maintained	Yes	Yes	Contact Tracing with iBeacon signals
Corona Trace	Small scale social distancing notification	Android, iOS	Organization has no public mer	No	Yes	anonymous location sharing to notify users
NextTrace	Contact tracing for regular people	unclear	NextTrace Team	No	No	curb COVID-19 transmission
Pan-European Privacy-Preserving Proximity Tracing	Contact tracing for specific group	no link provided	no link provided			no link provided
Contact Tracing app	Small scale social distancing notification	Web	Contact Tracing	Yes	Yes	Remember who you met to slow spread of covid
Contact Tracer	Database	Android, iOS	Contact Tracer Team	Yes	Yes	Recognise and connect to other devices no matter the platform
Private Tracer	Disseminating Information	Android, iOS	(Official?) Dutch Project	Yes	Yes	
Covi-id	Contact tracing for regular people	Android, iOS		No	No	Covid-19 risk management
Contact Tracing Covid19	Disseminating Information					
KeepDistance	Small scale social distancing notification					
Hansel	Contact tracing for regular people					slow the spread of covid-19
C19X	Contact tracing for regular people	Android, iOS	C19X	Yes	Yes	Secure Contact Tracing App
Corona-Warn-App	Contact Tracing App for specific group	Android, iOS	SAP and Deutsche Telekom	Yes	Yes	Germany's Covid Exposure App
Pandoa (Not Maintained refer to Safe Paths)	Contact tracing for regular people	Android, iOS	Not Maintained	No	Yes	Contact Tracing App from Hackathon
Ito	Contact tracing for regular people	Android, iOS	TCN Coalition	Yes	Yes	Privacy First Contact Tracing App
Path Check/Safe Paths	Contact tracing for regular people	Android, iOS	Path Check	Yes	Yes	Contact Tracing App
CovidWatch	Contact tracing for regular people	Android, iOS	Covid Watch	Yes	Yes	anonymous exposure notification app
CoEpi	Contact tracing for regular people	Android, iOS	CoEpi	No	Yes	Anonymous Contact Tracing
Hamagen	Contact Tracing App for specific group	Android, iOS	Israel's Minestry of Health	No	Yes	Israel Contact Tracing App
Global Epidemic Prevention Platform	Disseminating Information	Android	Ghana Health Service	Yes	No	Prevent spread of transmissible infectios diseases
Mind the Gap	Small scale social distancing notification	Android, iOS	Hack Partners	Yes	No	Office Social Distancing
covid-19 contact tracing	Database	Web	??	No	Yes	Consolidate and Operationalize data collected
Enigma SafeTrace	Database	Web	Enigma team	No	Yes	Centralize and protect the data collected
covid alert	Contact tracing app for regular people	Android, IOS	OpenMinded	Yes	Yes	A privacy-preserving app for comparing last-known locations of coronavirus patient
CovidSafe	Contact tracing app for regular people and Disseminating informatic	Android, IOS	??	Yes	Yes	slow the spread of covid-19
Covid Community Alert	Contact tracing app for regular people	Android, IOS	??	Yes	Yes	slow the spread of covid-19
Coalition	Contact tracing app for regular people	Android, IOS	??	Yes	No	slow the spread of covid-19
COVID Trace	Contact tracing app for regular people	Andrioid, iOS	??	Yes	No	slow the spread of covid-19
DP^3T	Contact tracing app for regular people	Android, IOS	International collective	Yes	Yes	protect civilians' privacy and slow the spread of covid-19
Open Covid Trace	Contact tracing app for regular people	Android, IOS	??	Yes	Yes	slow the spread of covid-19
CovidTracer	Contact tracing app for regular people	Android, IOS	??	Yes	Yes	slow the spread of covid-19

App Name (hyperlink to github or website)	Technologies	APIs Used	Data Collected	Summary (one sentence)
my app	BLE, QR code			
Covid World	BLE	unclear	Bluetooth Beacon Data	Aiming at helping fight COVID-19 spread by collecting anonymous data about people meeting each other
Corona Trace	GPS	unclear	Location	Building pandemic response tools to help keep individuals and communities safe
NextTrace	GPS, BLE	Google Timeline	Location	A scalable survey-based digital contact\tracing platform to curb COVID-19 transmission
Pan-European Privacy-Preserving Proximity Tracing	BLE	no link provided	no link provided	no link provided
Contact Tracing app	none	unclear	Contacts	Notifies your direct and indirect contacts if you log that you show symptoms or test positive
Contact Tracer	BLE	unclear	Nearby people stored locally	The point of the app is being able to recognise and connect to other devices no matter the platform and exchange UUID's to record for how long people have been in contact
Private Tracer	BLE			Goal is to find out if a contact tracing app can be an effective tool in batting the SARS-CoV-2 virus
Covi-id	QR code	unclear	unclear	Covid-19 risk management designed to protect privacy
Contact Tracing Covid19	agnostic			
KeepDistance	BLE			
Hansel	GPS			
C19X	BLE, SHA	ENS	Bluetooth Beacon Data	App that enables anonomous and secure contact tracing on many devices
Corona-Warn-App	Bluetooth	ENS	Anonymous bluetooth ID data stored locally	An app that enables you to retrieve test results electronically, and it helps to identify possible exposures you have had to people diagnosed with COVID-19
Pandoa (Not Maintained refer to Safe Paths)	GPS		Location stored locally	Remove from list?
Ito	Bluetooth	???	Nearby people stored locally	Ito is a privacy-first contact tracing app to fight SARS-CoV-2 and other pathogens.
Path Check/Safe Paths	GPS	ENS	Location	COVID Safe Paths is a mobile app for digital contract tracing (DCT) sponsored by Path Check a nonprofit and developed by a growing global community of engineers, designers, and contributors.
CovidWatch	BLE	ENS	Anonymous bluetooth identifiers	Free and Anonymous Exposure Notification App (implemented at University of Arizona)
CoEpi	Bluetooth, GPS		Opt in to share bluetooth contacts and symptoms	A privacy-first system for anonymous Bluetooth proximity-based exposure alerting based on voluntary symptom sharing.
Hamagen	BLE		Local location and contact matching	Privacy ensured contact tracing app that informs people of possible exposure for Israel
Global Epidemic Prevention Platform	GPS?		Location, Phone Call Access	Informs Ghanian government of travel to epidemic prone areas and sends information to users in those locations
Mind the Gap	Bluetooth, Audio		None	Monthly subscription based mobile app that alerts users in office settings when they are not social distancing with a sound.
covid-19 contact tracing		None	Inputed data from frontend	Early prototype of backend for a web app. More concerned with how to maximize the use of data rather than how to collect it
Enigma SafeTrace		None	Inputed data from frontend	Privacy preserving Database-as-a-service
covid alert	GPS	unclear	GPS location	Contact tracing alert that was abandoned for contribution to anther project (Corona-trace)
CovidSafe	Bluetooth	ENS	location	A tool to alert you about highly relevant public health announcements, potential exposure to COVID-19, and to assist public health officials.
Covid Community Alert	Bluetooth	unclear	Anonymous bluetooth ID and optional location	Covid alert app
Coalition	Bluetooth	unclear	Anonymous bluetooth ID	Fully functioning app available for free
COVID Trace	Bluetooth, GPS	ENS	location and anonymous bluetooth ID	Fully functioning app available for free
DP^3T	Bluetooth	ENS	bluetooth id	Fully functioning app available for free
Open Covid Trace	BLE	unclear	unclear	Fully functioning app available for free
CovidTracer	Bluetooth	unclear	bluetooth id	Fully functioning app available for free

E. Beacon Library (RSSI vs. Distance in meters)

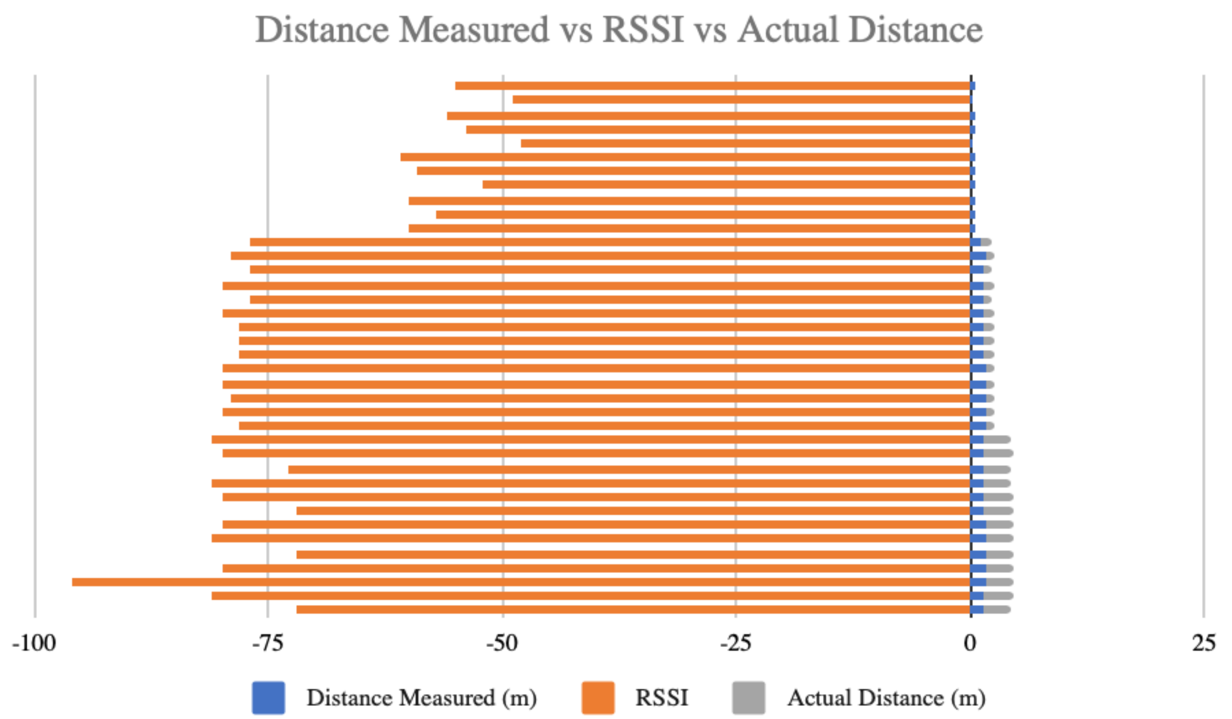
When we found the AltBeacon library during our research, we initially tested it to determine how accurately it could estimate distances. We tested the library by using it between smartphones at various distances between each other and collecting the actual distance, predicted distance, and RSSI. The graph below shows a plot of RSSI against actual distance.




RSSI vs. Time (unix millis)
RSSI
0
-25
-50
-75
-100
Time (unix millis)

F. Beacon Library (Distance in meters vs. RSSI vs. Actual Distance in meters)

The plot below shows the 'Distance' which is the value predicted by the AltBeacon library. This distance (in meters) is predicted by the model that the AltBeacon library includes.



G. Google Sheets Python Parser Library Requirements

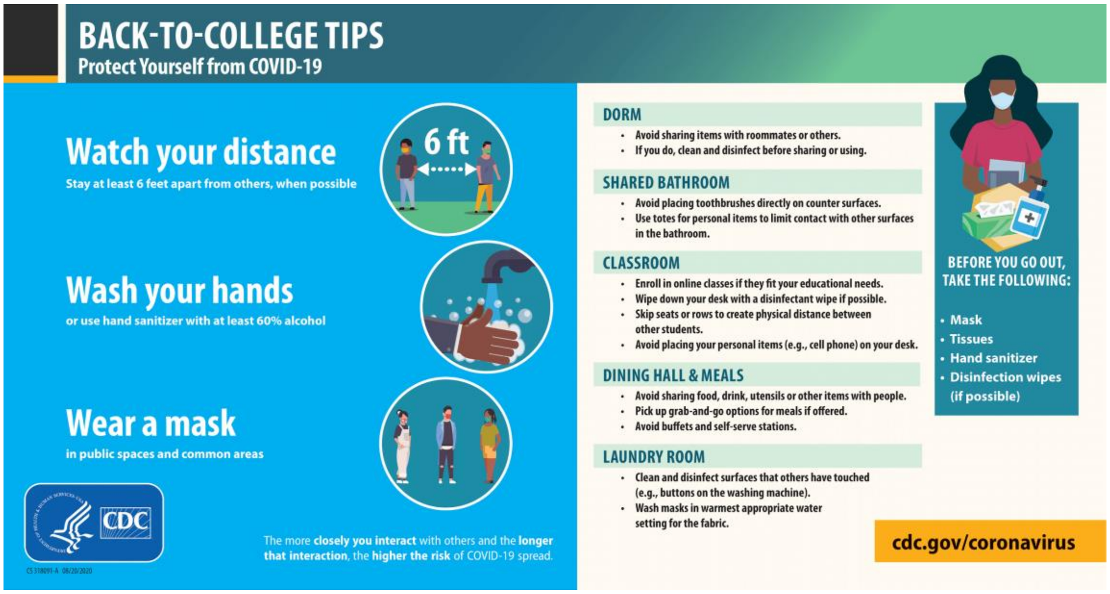
This is the list of Python packages used for the google sheets parser program.

```
google-api-core==1.22.3
google-api-python-client==1.12.5
google-auth==1.22.0
google-auth-httplib2==0.0.4
google-auth-oauthlib==0.4.1
google-pasta==0.2.0
googleapis-common-protos==1.52.0
oauth2client==4.1.3
oauthlib==3.1.0
mysql==0.0.2
mysqlclient==2.0.1
```

H. COVID-19 Infographics

I. Back to College Tips [10]

Infographic produced by the CDC with tips for reducing exposure to COVID-19 in a college setting.



J. COVID-19 Risk Index [16]

Infographic from COVID-19 RECoVERY CONSULTING showing the risk factors for different activities.

K. What you should know about COVID-19 to protect yourself and others [12]

Infographic from the CDC with tips and info about protecting yourself and others from COVID-19