Secure Location-Based Contact Tracing for COVID-19

Gokay Saldamli
Computer Software Engineering
San Jose State University
San Jose, CA 95192, USA

Jojo Joseph
Computer Software Engineering
San Jose State University
San Jose, CA 95192, USA

Mitra Gunakara Nayak
Computer Software Engineering
San Jose State University
San Jose, CA 95192, USA

Sidharth Jayaprakash
Computer Software Engineering
San Jose State University
San Jose, CA 95192, USA

Sushant Mathur
Computer Software Engineering
San Jose State University
San Jose, CA 95192, USA

Levent Ertaul
Department of Computer Science
California State University East Bay
Hayward, CA 94542, USA

***Abstract*—COVID-19 became one of the most fatal outbreak in the modern history. At the time of writing, about 150 million people were reported to be infected and more than 3 million people dead worldwide. It has caused a worldwide economic slowdown, with many businesses having to close and lay off their employees. It has also exerted mental stress among individuals as they worry about their family and loved ones.**

In order to bring things back to normal, it is absolutely necessary to control the spread of this disease. This can be done effectively using contact tracing, testing and isolating of individuals who could be exposed to the Coronavirus. Many countries have followed this approach and have been successful in limiting the spread of infections to an extent. This also helps health authorities in planning the roll out of vaccines and healthcare resources to areas with higher infection rates.

In this study, we attempt to create a contact tracing solution in a secure way. In our approach we try to inform people who have been in contact with an infected individual and have them quarantined and tested to avoid possible community spread. This is achieved through contact tracing using location data in a decentralised manner. The system informs the suitable group of people if an infection is reported but with complete anonymity. The system also provides a way to track the number of infections across various locations. The outcome will be better identification and control of community spread of this infectious disease.

***Index Terms*—COVID-19, contact tracking, Scalability, security**

I. Introduction

The harm inflicted by the COVID-19 pandemic is unprecedented and unfathomable. With the severity of disruption caused and the challenges faced, the pandemic can be called the most devastating healthcare crisis of the century. COVID-19 is an abbreviation for "Coronavirus disease 2019". The virus strain was first discovered in Wuhan, Hubei Province, China, in December 2019. Since the onset of the pandemic to the day the paper is written, 65,964,393 have been diagnosed with this infection, and 1,519,448 lives were lost globally [1]. Unfortunately, the United States has the highest fatality rate for COVID-19 amongst other developed nations [2].

As important as it is to find a cure and a vaccine for this infectious disease, it is equally essential to help regulate the disease's spread. One such critical practice is contact tracing. There are many solutions available in the market for contact tracing. However, our study aims to make a contact tracing application that is more secure and privacy-preserving. Before we discuss our system's working through this paper, it is necessary to throw some light on the key concepts and terminologies that drive this effort.

A. What is contact tracing?

Contact tracing dates back to the 1920s and has been used to control infectious diseases ever since. Contact tracing involves "identifying, assessing and managing people who have been exposed to an infectious disease" [3]. The main goal and motive of contact tracing are to break the possible chain of infections and its onward transmission. In simpler times, simple techniques like manual contact tracing were used. This was not the most effective way to break chain transmission as it was not a scalable solution. Nonetheless, with the technology boom and the invention of smartphones enabled with the internet and other technologies, the process has become efficient, scalable, and reliable. In this study, we discuss ways to implement a secure and stable contact tracing technique.

B. What is a GPS?

GPS stands for Global Positioning System. History tells us that sailors used to use stars and constellations to judge their current positions in the sea and navigate using the same. Thanks to technological advancements, individuals do not need to learn this art. We use GPS to direct, navigate, and locate us. GPS consists of about 30+ satellites orbiting the earth. With the help of these satellites, we can realize the exact latitude and the longitude on which we are present.

GPS works based using three major components [4]. The first and most vital component is the satellite. The satellites which take the place of constellations for positioning are expected to be at a specific position at a given time. The next component, called the ground stations, aid in satellite alignment and ensure the expected positions are maintained. They also keep information regarding the operational health of each of the satellites. The third component is the receiver. A receiver placed on each device knows its position by receiving the signals sent from the satellite. It makes use of at least four satellite signals to recognize it's position. Depending on the distance from each of the four broadcasting satellites, the exact location is calculated within 5 meters in case of smartphones.

II. Related Work

A. Bluetooth Solution by Apple and Google

Bluetooth is a wireless technology used to exchange data over short distances using Ultra High-Frequency radio waves. It operates between the 2.402 GHz - 2.480 GHz bands [5]. As Android and iOS mobile systems account for almost all of the mobile devices used worldwide, this ensures an extensive coverage of the population for contact tracing [6]. The device is assigned a random ID, which changes every 10-20 minutes. These devices then communicate with other nearby devices and exchange their IDs. Periodically it will check for IDs that have been marked as COVID-19 positive and warn the user that they have been in the proximity of someone who has tested positive. This solution maintains privacy as the ID is not related to the person, and the IDs frequently refresh hence not tying down one location to a single ID.

B. Wi-Fi Solutions

Wi-Fi is a set of wireless network protocols widely used for local area network connections and internet connectivity. This technology can be leveraged to fetch the Media Access Control (MAC) addresses of nearby devices and store it. MAC addresses are chosen as the identifier as they do not change for a device. The Received Signal Strength(RSSI) is the measure that can be used to estimate the distance of a given device from other devices. However, this solution is not a feasible approach and brings privacy concerns as the MAC address of every device coming under the range of a Wi-Fi network is stored.

C. GPS Solutions

Global Positioning System (GPS) is leveraged to create a location map of a user. GPS sensors in user's devices also provide much greater precision of location than Bluetooth or Wi-Fi alternatives. Hence, most of the existing covid tracing applications are making use of GPS to implement their tracing algorithms [7]. The geo-location data provided by the sensors in the user's devices can be collected in a JSON format, making it much easier to be used in the applications. This data is stored in a server and can be compared periodically to determine if a user has come in contact with an infected person. If it is found that a user has a positive contact, a notification is sent to the user device alerting the user about this. However, a faulty GPS sensor can lead to inaccurate location measurements, resulting in false positives.

D. Decentralized GPS Solutions

A decentralized GPS solution is one where the computation of potential proximity to an infected person is done in a distributed manner [8]. We can achieve this by sending periodic location information to mobile devices to check if any of these locations have been in proximity to itself in the past 14 days. This reduces the workload on the server and enables scaling horizontally. The user devices need to keep data of their locations for the past 14 days. Once they receive a positive person's location data, they have to check with their local data if see if they have come across them.

E. Centralized GPS Solutions

A centralized GPS solution is one where the computation of potential proximity to an infected person is done on a central server [8]. All the devices send their location data to a central server, and it computes if two people are in close enough proximity to spread COVID-19 from one person to another potentially. The server calculates and sends out a notification to a device if it concludes that the person has been close to someone who has tested positive for COVID-19. Scaling this method is hard as a central server is responsible for all computation. If the number of devices increases beyond a point, it becomes expensive to scale to accommodate these devices.

III. Problem Statement

Given the widespread disruption and impact caused by the Corona Virus, we need to limit its spread. One of the necessary and effective ways to combat the spread is via Contact Tracing. As discussed above contact tracing has been in use for a long time. But given the extent of spread and concerns regarding privacy concerns of the user, we need smart yet secure contact tracing methodology.
Our problem statement is to design and implement an efficient, scalable and secure framework for contact tracing. This application would help us in curbing and regulating the community spread by notifying users if they have come in contact with any infected individual. With this application the user will be able to receive the following services.

- Self Report
  We plan to deliver an android application for individual use to self report himself/herself while ensuring privacy.
- Notification
  The user will receive notifications if he or she has been in the vicinity of an infected individual.
- Visualisation
  The admin will able to see through a web application the current trend of infection and vaccination both state wise and date wise.
- Vaccination spots
  An admin will be able to upload vaccination points which the end user will be able to access.

IV. System Design

A. Architecture

The system follows a decentralized approach and can be divided into three parts. The client application which is deployed in the mobile device of users, and the server to which the users send and receive data to or from, and the web interface for the administrators. A high level architecture of the system is shown in figure 1. The users would interact with the system via the application installed on their mobile devices. The application can be run in android devices. It collects data in background and stores it in the device in the SQLite database which is present in the device. The administrator uses the web application written in React JS to interact with the system and visualizations present to him. Both the mobile device and the web application interact with the node js backend server which provides REST API endpoints for various functionalities. It also stores data sent from users, in this case the self reported infected points as well as the number of infections in Mongo database.

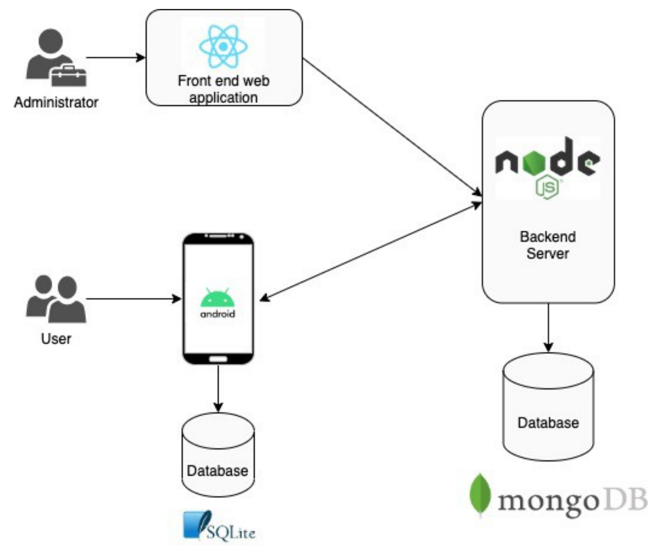

Fig. 1. High level architecture of the system

B. Description of Components

1) Mobile Application: The mobile application collects the users location from the background every 15 minutes. This collected data is stored in the users device as rows in SQLite database along with the timestamp. No other fields are collected by the application. The mobile application also collects the data for infected points from the server every day. This data is checked with the points collected at the users device. The user is notified if he was within 20 meters of the points which were reported as infected by the server. The user can also report himself as infected, at which point his location data can be uploaded to the server for the previous day. The user can also clear his location data if he wishes to do so.

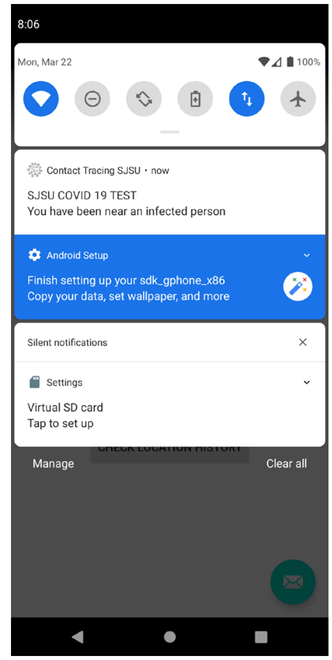

Fig. 2. Notification of Infection

Algorithm: The location data of infected users is fetched from the server every day at midnight. The application checks with it's own location history and calculates the distances between the two and ascertains if the user has been within 20 meters of an infected patient. If the user has, the application sends a notification alert.

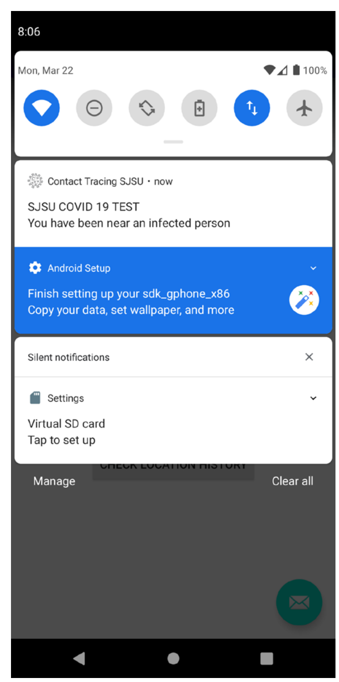

Fig. 3. Notification of Infection

Haversine Formula: This is the formula used to calculate the distance between two coordinates. It calculates the shortest line distance between two points on the Earth. According to the formula:

$$a = sin^2(\Delta\varphi/2) + cos(\varphi_1).cos(\varphi_2).sin^2(\Delta\lambda/2)$$

$$c = 2.atan2(\sqrt{a}.\sqrt{(1-a)})$$

$$d = R.c$$

Where φ is latitude, λ is longitude, R is Earth's radius (6,371 KM). All angles are in Radians.

Var Name	Type	Access
deviceId	long	public
latitude	double	public
longitude	double	public
date	long	public

TABLE I
LOCATION TABLE

If the pulled infected locations are close enough to the device's location according to the above formula, the user will be notified of them being in close proximity to an infected person.

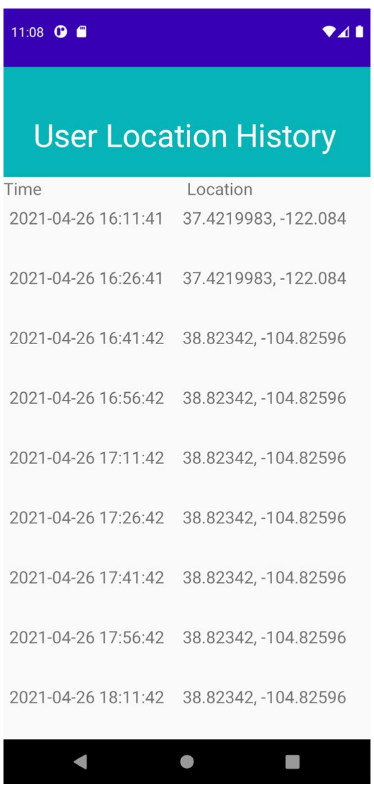

Fig. 4. location history

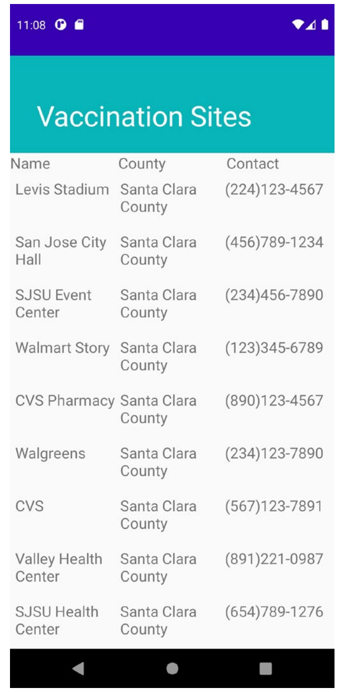

Fig. 5. Vaccination Centers

Database: The Mobile Application internally uses a SQLite database and we have used the Android Room APIs to interface with it. The location table will store the users location for the past 14 days on the device.

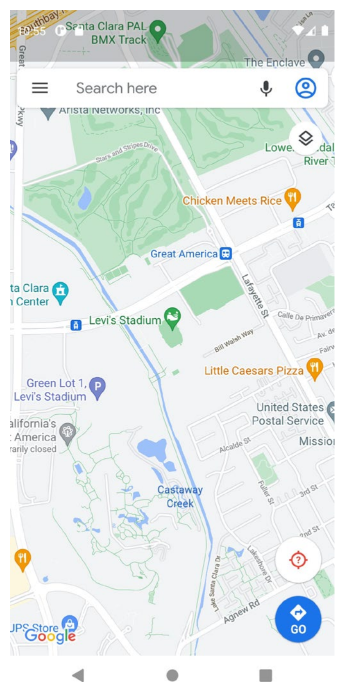

Fig. 6. OnClick Vaccination List

Background Tasks: There are two unique background tasks that execute on the android device of the user at all times.

- LocationGatherAndStoreWorker: Gathers the device's location every 15 minutes and stores it in the SQLite database.

- PullDataAndCheckWorker: Pulls data once every 24 hours and pulls infected points data and checks with the internal database to check if the device was in close proximity to any of those points.

Nearby Vaccination Sites: The user can check nearby vaccination sites based on their present location. The data is pulled from the backend and displayed in the Android application. Clicking on the vaccination point will open the location in Google Maps so that the user can navigate to the location.

2) Web Application: The web application provides the real-time visualizations in a map for the reported infected points as show in Fig. 7.

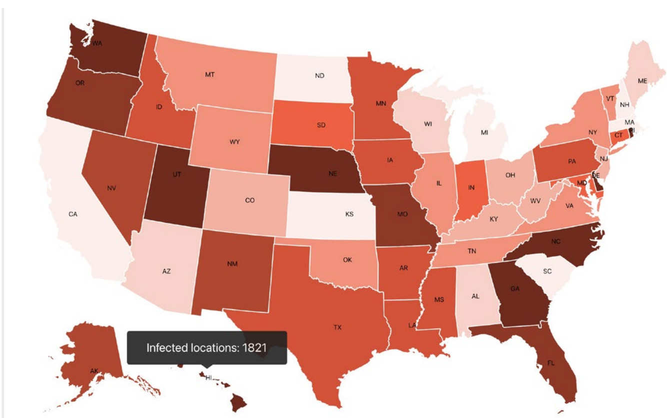

Fig. 7. Statewise Infected points

The administrator of the system can also view the number of infections which were reported by the users in the past 30 days and predicted values for the next 5 days as shown in Fig. 8.

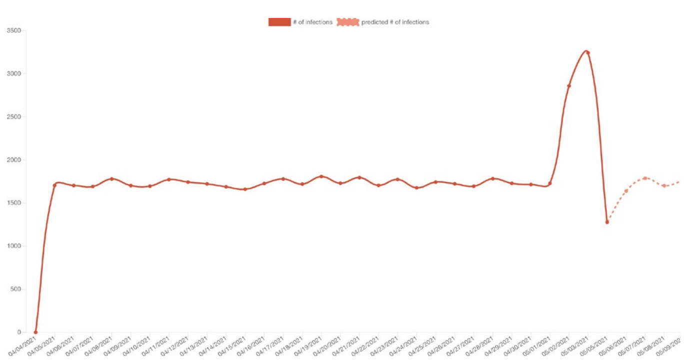

Fig. 8. Infection Trend

Moreover, an admin can also view the same metrics for vaccinations as shown in Fig. 9 and Fig. 10.

Also, an admin can add vaccination points which can be viewed by the user in the mobile application. Providing access to local healthcare agencies would help them in visualizing

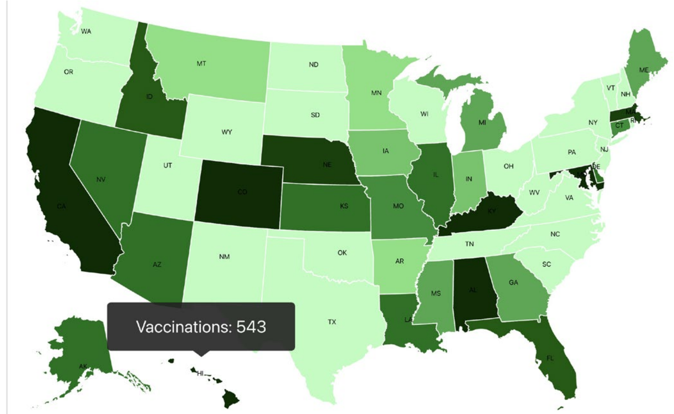

Fig. 9. Statewise Vaccinations

the infection as well as vaccination trends assisting them in taking appropriate measures to reduce the infection rate. The web application communicates with the backend server via REST APIs. The application is built with the react and the framework used to display the data visualizations is chart.js.

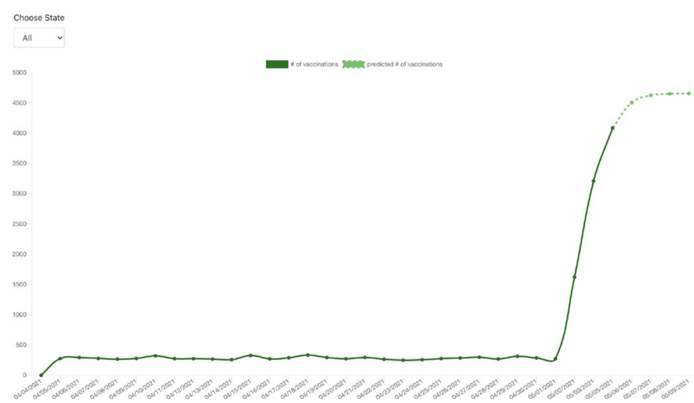

Fig. 10. Vaccination Trend

3) Backend Server: The backend server provides all the necessary REST endpoints for the system. The clients request the infected data points for a day using a REST endpoint. This data is returned to the user in JSON format. It can be used by a user to report himself as infected, for which a count is maintained in the Mongo database. The user can also upload his data to the server through REST api endpoint. It also facilitates the actions of the administrator such as providing data to support the visualizations by the web application component. The backend server provides endpoints to power visualisations such as infection rate, vaccinate rates etc across various states in the US. The infection rates and vaccination rates can also be extrapolated using techniques such as long short-term memory. This simple insight can help health workers and authorities to prepare for upcoming scenarios. It also provides endpoints to maintain vaccine locations, where users can go get vaccinated.

Database: The backend makes use of MongoDB a NoSQL database to store the details regarding infected locations and their timelines. The schema for the same can be in table II. The database also derives the geo location data such as he state, city, county etc from the latitude and longitude data. This helps in applying filters across the data to view individual state status. The vaccination data and infection data is maintained as documents in a collection. There is also provision to save vaccination locations, which is served to the mobile application, to be visible to the user.

Var Name	**Type**
latitude	Number
longitude	Number
timestamp	Date
reportedOn	Date

TABLE II
INFECTED LOCATION TABLE

V. EVALUATION METHODOLOGY

A. Functional and UI Testing

The system was evaluated based on the functionalities that were created. The specific use cases were tested as shown in table III. There were no issues found in the working of the functionalities.

Test Case	Expected Output
Admin should be able to login successfully to the dashboard with a valid username and password	Successful login
Admin should be to view different metrics in a numerical and graphical format	Application should successfully display tables and graphs of different metrics
User installs Android Application for the first time on their device	Introduction, tutorial shown to user and relevant permissions asked
Android device is powered on and functioning in a normal state	Application periodically collects location data in local database
User should be able to clear his location history	Application successfully deletes the user location history
Android application is installed and pulls data with a positive patient having been in their vicinity in the past 14 days	The device alerts the user that they may have been in contact with an infected person
The user should be able to self report himself or herself if infected	Acknowledgement of the reporting should be received by the user

TABLE III
FUNCTIONAL AND UI TESTS

B. Unit Tests

The unit tests in table IV were implemented in the android application and used to test the correctness of the system.

Test Case	Result
addLocationToDatabase()	Pass
deleteLocationFromDatabase()	Pass
testPositiveInfection()	Pass
testNegativeInfection()	Pass
ReportInfection()	Pass

TABLE IV UNIT TESTS.

C. Testing various scenarios

1) Both users not infected: In this scenario, two users had the application installed in their devices and had close interactions. This interaction was recorded in their devices. Then the client devices pull data from server, and both the users were not near any of the data points sent from the server. In the scenario, it was expected that there would be no warning sent to both the users.

2) One user self reports as infected: In this scenario involving two users, after their close interaction, one user reports himself as infected. The second users device pulls data from the server, and finds out that he had interacted with an infected user. The second user is shown a notification of potential infection.

VI. RESULTS

We successfully created a system of applications which when working in tandem, manage to function as a replacement for traditional methods of contact tracing. Leveraging the fact that almost everyone uses mobile devices these days we were able to track and notify users when they have been in contact with someone who is infected with COVID-19. All while maintaining the privacy of the users and not needing to use any personal information of them with the exception of their location. We used readily available technologies and targeted the Android OS first as it accounts for over 70 percent of all the mobile devices used worldwide. Android also has the advantage of being a lot more widespread in poorer countries which are generally more densely populated. Additionally, the web dashboard enables admin users to get real-time insights on the current infection and vaccination metrics assisting them in taking informed decisions. Also, the ability to add and update vaccine point details through the dashboard helps the users to get the latest information regarding the vaccination sites available near their locations.

VII. CHALLENGES

A. Mobile Device Location Accuracy and Battery Drain

In the Android OS SDK, there are two types of location data that we can access as developers, fine location and coarse location. Fine location uses the GPS as well as triangulation methods to retrieve a users location. This results in better accuracy at the cost of using more power. Coarse location methods do not use the GPS hence saving power. We need to have a mechanism such that we get reasonably accurate location without draining the users battery by a large margin. One method to do this is using the last known location of the device, this will use practically no power and is the least intrusive. However, this location can be stale and we may miss out on a key data point of the users device. For our use case we have chosen to compromise and use more power and get better accuracy as we are trying to implement contact tracing for which we need the most accurate data that we can possibly get.

B. Relying on User Self Reporting Mechanism

To maintain privacy of the users we are not tracking any information specific to them. This raises the requirement that in case an user tests positive, it is their responsibility to report themselves as infected through the android application. Any instance where an user fails to report themselves in case of an infection would compromise the whole system and its functionalities. Hence we rely on the user to report an infection and this is human aspect we must face.

C. Scalability for Large Number of Users

Since we have designed a system in which the backend server would be computing the affected radius when an user marks himself as infected, it puts a very heavy load on the server. As the number of users increase, the probability of number of infected people would also increase substantially. This increase would require the backend server to be very scalable. Since we have developing and testing our system with a limited set of data, a load testing in such a large scale is not feasible. Hence, scalability of the system when the number of users increase is a challenge and limitation faced by our system.

VIII. Conclusion and Future Work

Secure Location-Based Contact Tracing for COVID-19 is an efficient yet secure contact tracing engine. By interacting with the mobile application the user will be able to self report himself or herself and thereby protect hundreds of others from being a super-spreader. Thus the application helps in breaking the chain and hence curtailing the speed of the spread of the Covid-19 virus. This engine can act as a medium or a tool in bringing the pandemic under control.

As a future work, we plan to publish our mobile application on to the Google Play store so that it is available to the general public. Also, we plan to bring out an IoS version of the same application and make it available on Apple store. We also plan to add more visualizations for infection and vaccination trends helping the admin users get more real-time insights.

References

[1] "Worldometer covid-19 coronavirus pandemic," Available at https://www.worldometers.info/coronavirus/ (12/04/2020).

[2] M. F. Cortez, "U.s. has highest covid-19 death rate in developed world," Available at https://www.emsworld.com/news/1225033/us-has-highest-covid-19-death-rate-developed-world (10/12/2020).

[3] "Coronavirus disease (covid-19): Contact tracing," Available at https://www.who.int/news-room/q-a-detail/coronavirus-disease-covid-19-contact-tracing (07/28/2020).

[4] "How does gps work?" Available at https://spaceplace.nasa.gov/gps/en/ (06/27/2019).

[5] "Bluetooth," Available at https://en.wikipedia.org/wiki/Bluetooth (06/27/2019).

[6] K. Michael and R. Abbas, "Behind covid-19 contact trace apps: The google–apple partnership," *IEEE Consumer Electronics Magazine*, vol. 9, no. 5, pp. 71–76, 2020.

[7] E. Hernández-Orallo, P. Manzoni, C. T. Calafate, and J. Cano, "Evaluating how smartphone contact tracing technology can reduce the spread of infectious diseases: The case of covid-19," *IEEE Access*, vol. 8, pp. 99 083–99 097, 2020.

[8] N. Ahmed, R. A. Michelin, W. Xue, S. Ruj, R. Malaney, S. S. Kanhere, A. Seneviratne, W. Hu, H. Janicke, and S. K. Jha, "A survey of covid-19 contact tracing apps," *IEEE Access*, vol. 8, pp. 134 577–134 601, 2020.