Contact tracking for Covid-19

***Abstract*—COVID was a urgent issue for human society and one way to fight with it was to identify and test the close contacts of COVID patients efficiently. An android application that utilized bluetooth and GPS was developed. Results showed that it allowed patients to report their COVID infection anonymously and people can check whether they are close contacts of these patients efficiently.**

***Index Terms*—COVID, Contact, Android, Bluetooth, GPS**

I. Introduction

A. Motivation

COVID is a serious and urgent issue for oru society. Before vaccine for covid is available for public, a key measure to control the spread of covid virus is tracking contacts of infected victim and test them [1]. However, it is still challenging to keep track of close contacts especially when the infected patient has complex trajectories. An example is that when people commute to workplace in peak hours by subway, they will contact so many people that tracking his/her close contacts will become extremely difficulty.

A traditional approach to address this is to publicize the trajectories of the infected people in media so that people who have visited these places will be informed and take covid tests [1]–[5]. However, some people may not notice such information, and fail to take the necessary covid test. This increases infection risks for the whole society. On the other hand, some people may be overworried about this information and take the unnecessary covid test, because they may visit these locations at some points but they actually do not have close contact with the confirmed case. These unnecessary covid tests will waste valuable medical resources.

There has been smarter solution to tackle this, and several APPs in android App store are available for downloading [2]–[5]. These App usually uses bluetooth to detect and record the close contacts(represented by bluetooth ids) of the mobile phone user [2]–[5]. If this mobile phone user is tested to be COVID positive, he/she can choose report this result by this app to the remote database(server) managed by a medical team [2]. In this process, the user's close contacts(bluetooth ids) stored in his mobile phone will be uploaded to this server which will notify the mobile phones corresponding to these bluetooth ids. This method is efficient and protect privacy quite well [2]–[5]. Its main disadvantage is that it can not classify the infection risk, because it can only detect close contacts within two meters. For people whose social distance with the infected victim is larger than 2 meters but may be less than 10 meters, they will not be informed. However, although these people may not need to take covid test, for our society safety, these people should be informed so that they can take some precautious measures(e.g., self isolation for 14 days) to avoid the spread of COVID virus.

This paper proposes a solution to enhance the existing Apps. Bluetooth is combined with GPS to jointly detect the close contacts. So, our app will not only collect the bluetooth ids within 2 meters but also record the mobile phone user's real time position by GPS. By this, it is expected that the risk level(determined by social distance) will be classified and people can make better decision based on their risk levels. Additionally, although it is not our proposed object when the project starts, gyroscope is tried in our project to detect orientation and classify the risk level furtherly. This makes sense, because people who talk to each other face to face will have higher infection risk than people standing back to back. The details for the risk level classification will be discussed in section II system architecture.

B. Aims and Objectives

This project aims to help detect close contacts of COVID-19 efficiently and precisely. The objectives are as follows:

- Use bluetooth the detect the close contacts whose social distance is less than 2 meters.
- Use GPS to record the real time positions of App users and use the records to calculate social distance for risk levels classification.
- Use gyroscope to detect orientation for risk classification**(not our proposed object, but tried)**
- Develop database that simulate the remote server to store and analyze data.
- Develop user friendly user interface(UI)

As commercial APP in the APP store mainly focuses more on the sensor data collection and the data analysis work is usually done by a remotely separate database managed by a medical team, our project also has more focus on the data collection by the APP. Once the collect data can be uploaded to the database, the main work of our project is completed.

II. System Architecture

The system architecture is shown in Fig.1. This architecture mainly include a database in personal computer and the APP installed in smart phone. Their basic function is data collection for APP and data analysis for database.

In the APP, bluetooth, GPS and Gyroscope are wrapped by three threads so that each sensor has time opportunities to collect data: bluetooth device name and signal strength for bluetooth, longitude and latitude for GPS and angular velocity for Gyroscope.

The sensor data will be collected automatically once the user open the app. For GPS and gyroscope, they will write

their collected data to files in mobile phone every 1 second. 1 second time interval is small enough to accurately record human's position and orientation. For Bluetooth, it will write its the data to the file immediately after the data is collected. Otherwise, the bluetooth may miss some data that are not recorded timely.

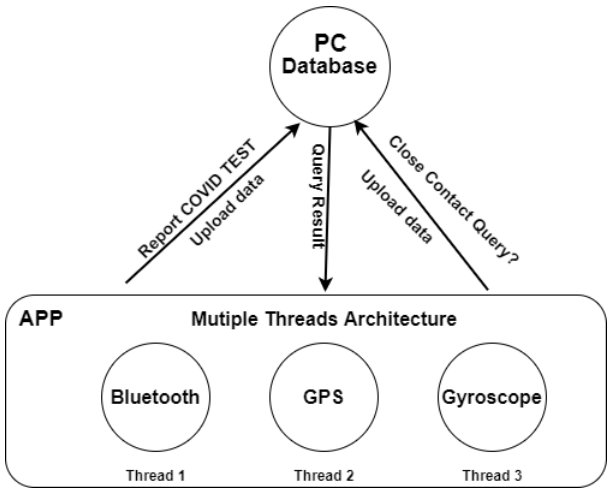


Fig. 1. Overal system architecture.

When the user is tested to be COVID positive, he can choose to report the test result to the database(server) by clicking the button 'REPORT TEST RESULT' in the UI in Fig.2. By this, the APP will upload this user's data files(that store collected bluetooth ids, his/her GPS data etc) to the database as shown in Fig.1.

An user can also choose to submit his queries about his infection risk by clicking the button 'NOTIFICATION FOR INFECTION RISK' in Fig.2. By this, the APP will also upload the user's data file to the PC database which will calculate the risk level and send the result back to the APP. For this project, we focus more on the APP itself and risk level classification is usually done by the database instead of the APP. Therefore, the risk level classification algorithm is only described and analyzed here in the report.

The procedure for the risk level calculation is as follows. Once receive queries, The database will firstly use the user's bluetooth ids to match all the collected bluetooth ids uploaded by the infected patient. If matched, the database will inform the APP that its user has very high infection risk. If not matched, the database will use GPS records to calculate the least distance between the user and all patients. If the distance is less than 10 meters at some time, the database will inform APP that the user has high infection risk. Otherwise, the APP will receive notification from database that the user has minimum infection risk. The data from gyroscope can be used for further classification, but these data are not used for some reasons which will be explained later.

III. Technical Approach

This section introduces how the application is implemented in android studio.

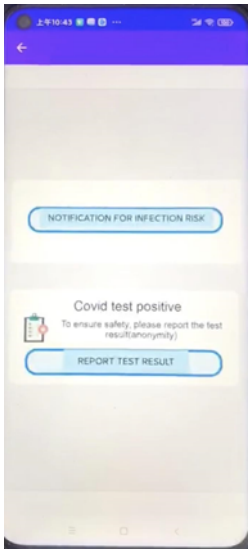


Fig. 2. The user interface for queries and reporting.

A. Multiple Threads Architecture

As shown in in Fig.1, three work threads are used in this study: gyroscope Thread, gpsThread and bluetoothThread. Each thread has roughly the same priority.

To implement multiple threads, two classes(Handler and HandlerThread) are used. Handlerthread(that inherits from thread) owns a looper that can manage message queue. After the HandlerThread starts, it can get its looper and use this looper to instance handlers(gps Handler, gyroscopeHandler or bluetoothHandler) that can push or receive messages from the message queue. The relationship between Handler and HandlerThread is shown in Fig.3.

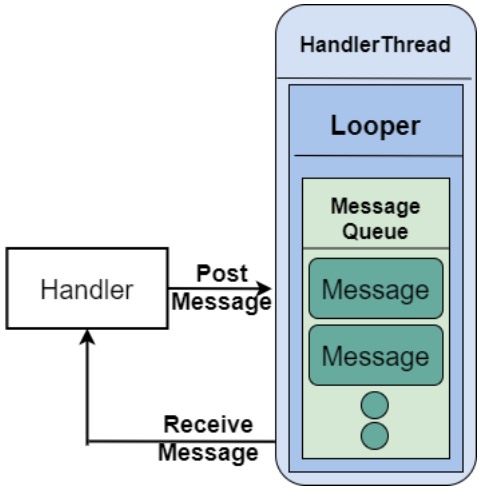


Fig. 3. Handler and Handlerthread

The pseudo-code for gps,gyroscope and bluetooth thread is shown in algorithm 1,algorithm 2,algorithm 3

Algorithm 1 Worker Thread for GPS

```
gpsThread = new HandlerThread()
gpsThread.start()
gpsHandler = new Handler(gpsThread.getLooper())
gpsHandler.post(newRunable()
{
    getLocation(); %Get GPS data every 1 second
}
```

Algorithm 2 Worker Thread for Gyroscope

```
sensormanager = GetSensorService
gyroscopeThread = new HandlerThread()
gyroscopeThread.start()
gyroscopeHandler = new Handler(getLooper())
gyroscopeHandler.post(newRunable()
{
    onSensorChanged(); %Get gyroscope data
}
```

B. GPS

For GPS, a class "GPStracker"(that implements LocationListener) is created and the core method "getLocation"(that returns the current longitude and latitude values) is developed. The getLocation function is then called every 1 second in the gpsThread shown in algorithm 1.

For the method "getLocation", it will firstly check whether relevant permission is granted and GPS is enabled. If these conditions are met, it will request location update via location manager from GPS receiver. The pseudo-code for "getLocation" is shown in algorithm 4.

After the longitude and latitude values are acquired regularly, they are used to estimate the distance d between any two locations A and B. In this study, haversine formula(that treats the earth as a sphere and calculate the distance between the two points along a great circle of the sphere) is adopted.

$$d = 2r \arcsin(\sqrt{\sin^2(\frac{\phi_B - \phi_A}{2}) + \cos(\phi_A)\cos(\phi_B)\overline{\sin^2(\frac{\lambda_B - \lambda_A}{2})}}) \quad (1)$$

Where r is the earth radius. ϕ_A, ϕ_B represents latitudes of location A and B respectively, and λ_A, λ_B are the longitude of location A and B.

Algorithm 3 Worker Thread for Bluetooth

```
bluetoothThread = new HandlerThread()
bluetoothThread.start()
bluetoothHandler = new Handler(getLooper())
bluetoothHandler.post(newRunable()
{
    enableDisableBT(); %Get bluetooth data
}
```

Algorithm 4 Get GPS Location

```
LocationManager lm = GetLocationService
if Permmision is granted  GPS is enabled then
    lm.RequestLocationUpdates()
else
    print("please enabble GPS")
end if
```

C. Gyroscope

The gyroscope data is collected via the sensor manager which can get the gyroscope sensor service from the mobile phone. Then the sensor manager will register the sensor event listener that will implement the function onSensorChanged(). This function will acquire the gyroscope sensor data once new gyroscope data comes.

Approximately, the app can get about 10 gyroscope data per 1 second. However the data is quite noisy, so these 10 data are taken average to reduce noisy. The pseudo-code for gyroscope is shown in algorithm 5.

Algorithm 5 Get Gyroscope Data

```
sensormanager = GetGyroscopeService
sensormanager.registerListener(sensorEventListener)
 onSensorChanged(event) {
 gyroscope_values = event.values;}
```

It turns out that gyroscope data quite inaccurate to capture the people's orientation. Therefore, although it has been implemented and gyroscope data has been collected in this project, they are not used in the end.

D. Bluetooth

To make the Bluetooth functions works properly, three different bluetooth broadcast receivers are set up in our app. Setting multiple BT broadcast receivers is important when a pair process and connection process is required, because it can prevent the null-pointer error when there is no device need to connection. Our Bluetooth function has a well-designed function of setting up pair and connection at the beginning, however, the users' device name are no longer required to be transferred by bluetooth function, so the connection and pair model are cut off from our project. The bluetoooth architecture is shown in Fig.14.

Another important problem for bluetooth in this project is how the bluetooth can determine the social distance between people. As people all bring their mobile phone which has its own bluetooth device name, the received Signal Strength Indictor(RSSI) signal is used in this study to judge whether people are within 2 meters social distance. This means Only the device name with the preferred RSSI signal value are treated as being within 2 meters distance and will be uploaded to the database for matching and analysis. The distance d can then be calculated by equation 2.

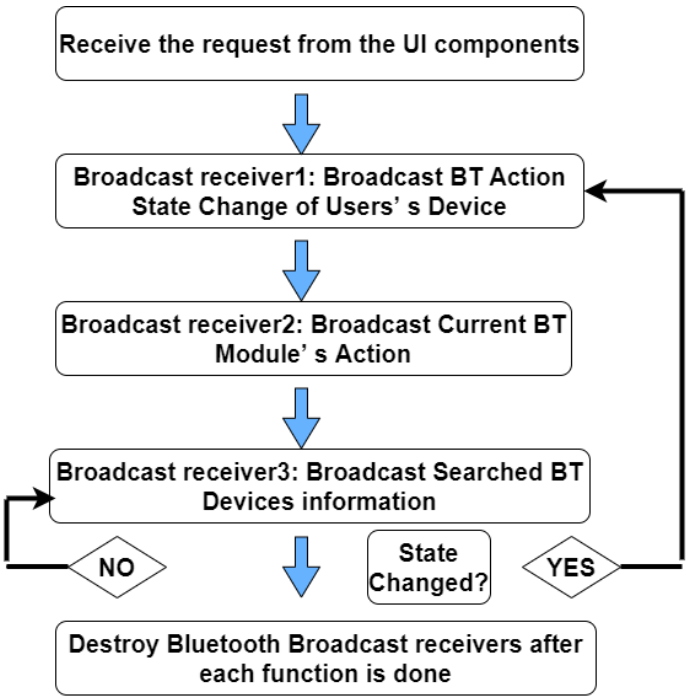


Fig. 4. Bluetooth Architecture

$$Distance = 10^{((MeasuredPower - RSSI)/(10*N))} \quad (2)$$

For example,if our measured power is -69 and the RSSI signal is -80, then the distance is $10^{((-69-(-80)/(10*2))} = 3.54\ meters$.

User also needs to agree that the bluetooth can be run by this app. So, Permission check is necessary. To do this,the fine and coarse permissions commands should be added into the manifest. The Bluetooth permission check function must also be in the same JAVA class where it will be called, or the permission check function will not be called. Additionally, the SDK version must be more than 21 to run the permission check API in the app.

E. Database

The database is used to store the received Bluetooth device name, MAC address, and RSSI signal strength, and send the result to the users. Three different databases are set in our application.

First one is that a txt file will be automatically created in our app, and the sensors reading will be stored into that txt file. Later, the values in these txt files will be uploaded to microsoft database. This can be shown in Figure .5

Second one is the Microsoft SQL server. App will upload the device name with the RSSI signal over the threshold to the database, and send the matched result to the user' s app. This is shown in Fig.6

Third one is the firebase realtime server. App will upload all the device name with RSSI signal to the firebase database, and show them to the user in real time. It is implemented by a build-in API named as firebase tool to show the user with searched bluetooth information. Similar to the MS sql server, the permission of server connection should be added into build.gradle(app name.app).

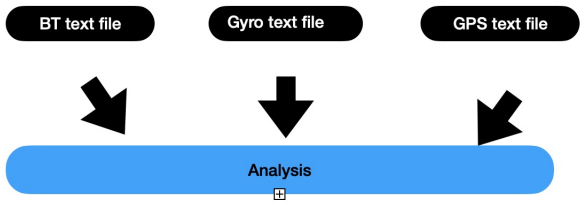


Fig. 5. Read sensor data from text files in Android

Fig. 6. Read sensor reading to database directly

F. User Interface

For User Interface(UI), the UI design from other similar commercial APP is referred and our UI mainly includes three parts.

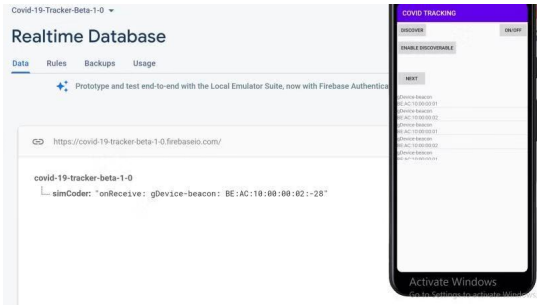


Fig. 7. firebase

The first part comprises of a welcoming page and the data privacy page.

The second part is to turn on the devices(bluetooth, GPS and gyroscope) by clicking buttons in the UI. When the user do this, the app will create and activate HandlerThread and Handler for these devices.

The last part(also shown in Fig.2) is to report test result and submit queries by interacting with the remote database.

IV. Experiments

This section introduces the experiment setup, results and the analysis. The tests for individual component are discussed firstly and then the experiment simulating the real environment is discussed.

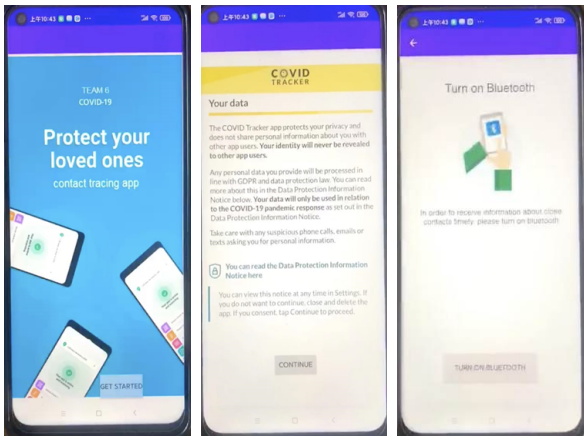
A. GPS

Compared to other approaches that relies on online map(e.g., google map) for GPS localization, the biggest advantage of our APP is that the GPS component can still work well without internet. However, it is uncertain about the GPS precision in our project. This is important because this determines to which degree we can trust the collected GPS value. Therefore, an important experiment for GPS is its precision test.

In this project, the GPS data from Baidu Map is assumed to have high localization precision and our collected GPS values(longitude and latitude) are compared to them. To do this, a 20 minutes' walk is taken around the residential house in a city in China and several landmarks(e.g., subway station, restaurant etc) near the home whose GPS value can be acquired from Baidu map are visited in this short journey. As shown in table II, the GPS data of our APP for these landmarks are then recorded and compared to the one from Baidu Map. The deviation of our GPS data from Baidu Map is about 0.009755% for longitude data and 0.010488% for latitude data. Although the comparison has proved that our GPS component can provide pretty satisfactory precision, Baidu Map only shows the GPS data for several distinctive landmarks and there are **NO** GPS data for many locations in this walking journey. Here, we offers an indirect method to prove that these locations also have good precision. Since in this short walking Journey, GPS data is collected by our APP every 1 second. By applying equation 1, the distance between consecutive GPS data points

TABLE I
User Interface

Welcome | Data Privacy | Turn on Bluetooth



Turn on Gyroscope | Turn on GPS | Database Interaction

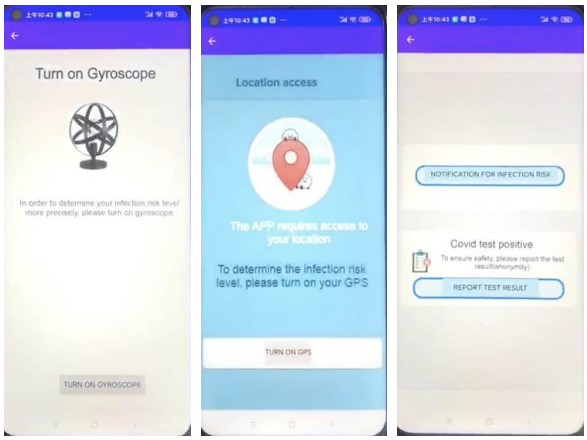


TABLE II
GPS Data Comparison among Baidu Map and Our APP

	Baidu Map		Our APP	
Location	Longitude	Latitude	Longitude	Latitude
Bus Station	115.838580	28.644705	115.824953	28.642082
House Gate	115.836545	28.644685	115.825592	28.642130
Store	115.836968	28.643544	115.825693	28.640890
Restaurant	115.837938	28.643746	115.826238	28.640533
Subway Station	115.836655	28.646876	115.824988	28.642900

can be estimated and so does the walking speed in this journey. As Fig.9 shows, the walking speed fluctuates around 1 m/s and the average walking speed is 1.218 m/s which is quite close to average human walking speed(1.4 m/s). This provides another evidence that our GPS component is quite precise for the whole 20 minutes' walk.

B. Bluetooth

To record the people pass by the app' s users in 2 meters, the bluetooth is used and user will receive the bluetooth ID,

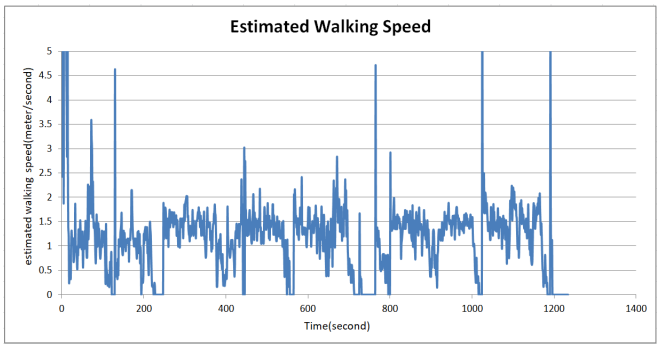


Fig. 8. The estimated walking speed by GPS

MAC address and RSSI signal strength through the bluetooth function.

When one opens the app initially, an bluetooth set up page will appear. User needs to run and shut down the bluetooth function module by clicking the button "ON/OFF". The permission check toast may be popped up to ask for the permission to run the bluetooth.

Then the user can hit the button "discover", and this will active the scan model which will scan the nearby devices name around 12 seconds.

Users may also want that his/her device can be detected by others so that other user can also acquire his bluetooth device name for close contact tracking. Our app has set up the button "enable discoverable" to realize this function.

Finally, developer may want to know the effect of this bluetooth setp. To show the effect, a textview in bluetooth set up page is created to show the information from the broadcast receiver(that will override the list-view in Bluetooth control panel dynamically). However, users may not want to see these information because they have no idea of their meaning and may even believe wrongly that something goes wrong. So, these information is just for app test purpose and will not shown in the Finale App.

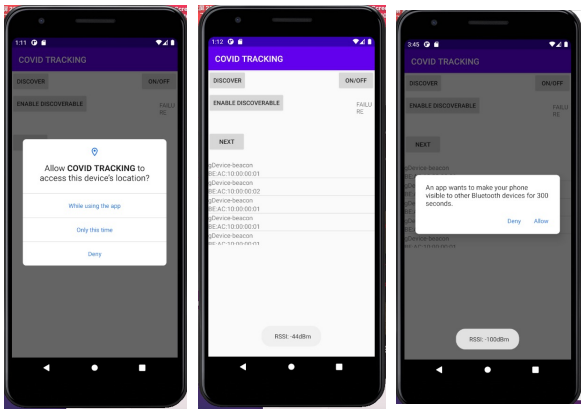
The effect of bluetooth setup is shown in Table.III.

The threshold of RSSI signal value is very important for the bluetooth to decide whether the social distance is within 2 meters. To get the threshold of the RSSI signal value of 2 meters' scale, a very long time experiment was undergoing in indoor environment. Because the signal strength will be affected by the temperature, the temperature is controlled in 76 Fahrenheit. Two devices are put on the same height objects, and collect around 1000 data with no objects between these two devices. After that, a volunteer sit between two devices as a barrier. The signal strength will reduce around 30% when 1000 data is collected. Finally, we acquire the mean value of the RSSI signal strength we collected when no barrier to decide the 2 meter's social distance. If the received RSSI signal value is higher than the threshold, the device will be considered as within 2 meters, and will be upload to server.

Some of the collected bluetooth data(shown in Android studio log) within 2 meters is presented in Fig.10. So, the bluetooth can work appropriately.

TABLE III
BLUETOOTH SETUP

Permission Checking	discover devices	enable discoverable



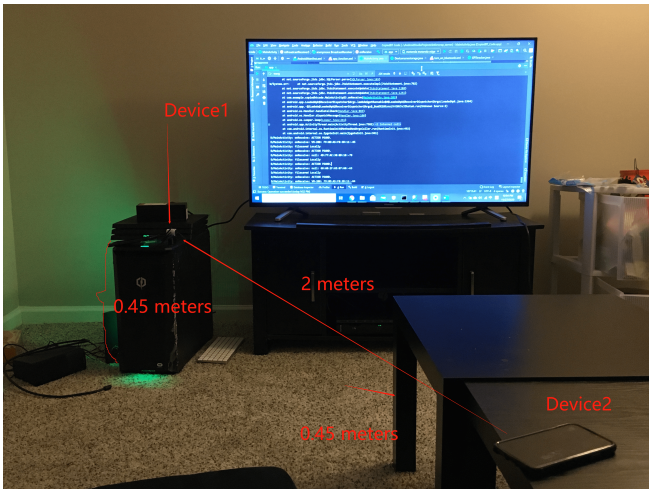


Fig. 9. Experiment Setup for RSSI test

C. Database

The results for database that read data from text file is shown in Fig.11,Fig.12,Fig.13. The sensor reading time has been converted into integer format.

D. Experiments in real environment

After each individual module has been developed, the developed APP can then be used to collect data in the real environment for further analysis in database(which has been described above). So, this subsection describes how data are collected in real environment. The data are stored in txt files in mobile phones and later will be uploaded to the database. The data collection algorithm is described in algorithm 6.

To test our app, a 20 minutes' walk is taken around a residential zone in a Chinese city. As our APP should work well in both the indoor and outdoor environment, most of the walk journey is in outdoor environment but several indoor locations(e.g.,convenience store) have also been visited

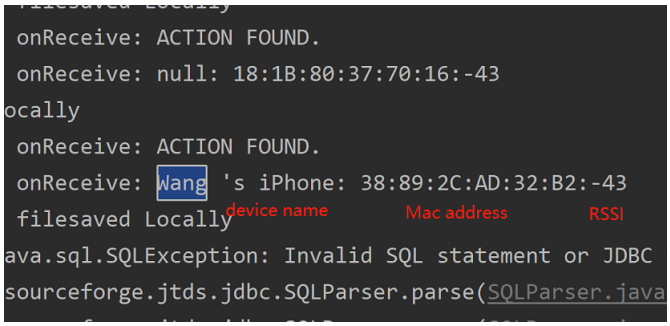
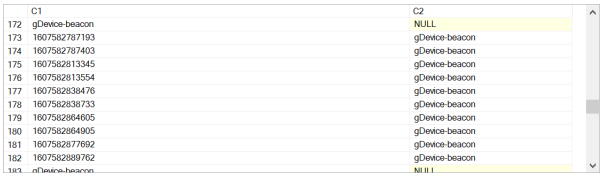
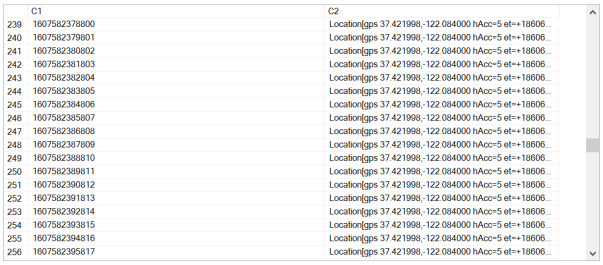


Fig. 10. The collected bluetooth information



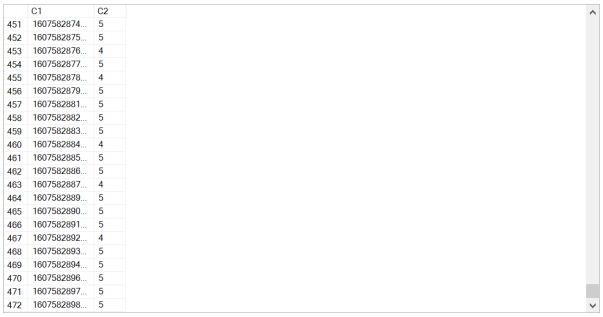
	C1	C2
172	gDevice-beacon	NULL
173	1607582787193	gDevice-beacon
174	1607582787403	gDevice-beacon
175	1607582813345	gDevice-beacon
176	1607582813554	gDevice-beacon
177	1607582838476	gDevice-beacon
178	1607582838733	gDevice-beacon
179	1607582864605	gDevice-beacon
180	1607582864905	gDevice-beacon
181	1607582877692	gDevice-beacon
182	1607582889762	gDevice-beacon

Fig. 11. Bluetooth



	C1	C2
239	1607582378800	Location[gps 37.421998,-122.084000 hAcc=5 et=+18606...
240	1607582379801	Location[gps 37.421998,-122.084000 hAcc=5 et=+18606...
241	1607582380802	Location[gps 37.421998,-122.084000 hAcc=5 et=+18606...
242	1607582381803	Location[gps 37.421998,-122.084000 hAcc=5 et=+18606...
243	1607582382804	Location[gps 37.421998,-122.084000 hAcc=5 et=+18606...
244	1607582383805	Location[gps 37.421998,-122.084000 hAcc=5 et=+18606...
245	1607582384806	Location[gps 37.421998,-122.084000 hAcc=5 et=+18606...
246	1607582385807	Location[gps 37.421998,-122.084000 hAcc=5 et=+18606...
247	1607582386808	Location[gps 37.421998,-122.084000 hAcc=5 et=+18606...
248	1607582387809	Location[gps 37.421998,-122.084000 hAcc=5 et=+18606...
249	1607582388810	Location[gps 37.421998,-122.084000 hAcc=5 et=+18606...
250	1607582389811	Location[gps 37.421998,-122.084000 hAcc=5 et=+18606...
251	1607582390812	Location[gps 37.421998,-122.084000 hAcc=5 et=+18606...
252	1607582391813	Location[gps 37.421998,-122.084000 hAcc=5 et=+18606...
253	1607582392814	Location[gps 37.421998,-122.084000 hAcc=5 et=+18606...
254	1607582393815	Location[gps 37.421998,-122.084000 hAcc=5 et=+18606...
255	1607582394816	Location[gps 37.421998,-122.084000 hAcc=5 et=+18606...
256	1607582395817	Location[gps 37.421998,-122.084000 hAcc=5 et=+18606...

Fig. 12. GPS



	C1	C2
451	1607582874...	5
452	1607582875...	5
453	1607582876...	4
454	1607582877...	5
455	1607582878...	4
456	1607582879...	5
457	1607582881...	5
458	1607582882...	5
459	1607582883...	5
460	1607582884...	4
461	1607582885...	5
462	1607582886...	5
463	1607582887...	4
464	1607582889...	5
465	1607582890...	5
466	1607582891...	5
467	1607582892...	4
468	1607582893...	5
469	1607582894...	5
470	1607582896...	5
471	1607582897...	5
472	1607582898...	5

Fig. 13. Gyroscope

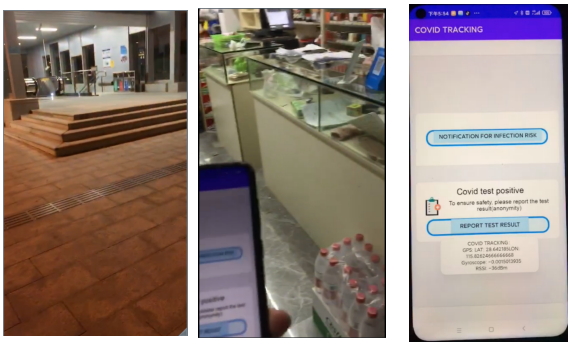
Algorithm 6 Data Collection

$Now = new\ Date();$
$SensorData = GetSensorData()$
$DataFile = newFile();$
$Writer = newFileWriter();$
$Writer.append(SensorData, Now);$
$Writer.close();$

to collect the data(GPS,gyroscope and Bluetooth) in indoor environment. Table.IV shows some of the outdoor and indoor locations that has been visited in this experiments. The APP interface during this experiment is also presented here. It can be noticed that in this interface page, the sensor values are shown in the toast in real time meaning that our three sensors can work together properly.

TABLE IV
OUTDOOR AND INDOOR EXPERIMENT

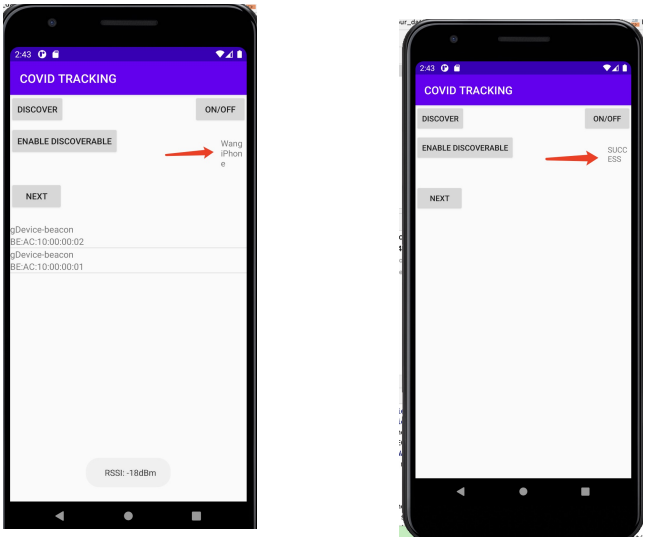
Subway station | Convenience Store | APP Interface



The collected data stored in the txt files are shown in Table.VI.

TABLE V
RECEIVE TWO METERS' BT DEVICE NAME EXPERIMENTARCHITECTURE

SQL Connection | Received Device ID



V. CONCLUSION AND FUTURE WORK

This project proposed an APP to track the close contact of the COVID patient efficiently and precisely.

The bluetooth component in the APP is developed and tested. Results has show that it can collect the bluetooth device information of the contacts within 2 meters and can not do it

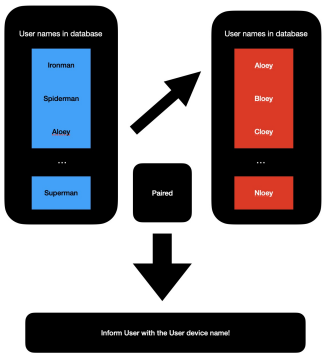
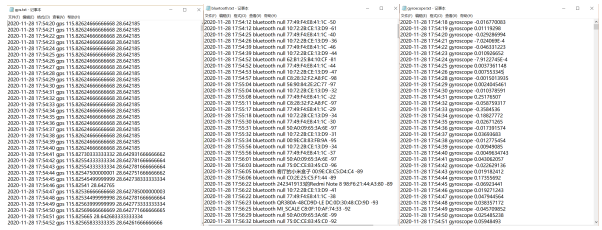


Fig. 14. The Communication of the Bluetooth and Server' s Architecture

TABLE VI
COLLECTED SENSOR DATA IN THE EXPERIMENT

GPS Data Bluetooth Data Gyroscope Data



beyond 2 meters, which has fully fulfilled one of the important objects in the project.

The GPS component has also been developed and tested in the project. Results shows that the GPS component has pretty high precision, meaning that it will be a very good complement to detect contacts beyond 2 meters but perhaps less than 10 meters. Therefore, another important object of this project has also been fulfilled.

This project has also developed the database to store and analyze the data. Results shows that the collected data in experiment can be uploaded to the database successfully, which lays good foundation for the further work.

There are several future direction for our project.

One is that our project has not incorporate the orientation to classify the risk levels because of the inaccurate gyroscope. In the future, work can be done to improve the accuracy of the gyroscope. Alternatively, one can also use other sensors for the orientation detection.

Another work is the power consumption issue. Both GPS and Bluetooth consume lots of power. So, it will be helpful that make the bluetooth and GPS module into sleep state when they are not necessary.

Finally, comparison should also be made with the commercial APP in the APP store. By this, the weakness and advantage of our APP will be more clear and future improvement can be made based on these comparison.

REFERENCES

[1] World Health Organization, "Contact tracing in the context of COVID-19," interim guidance, pp. 1–7,May 2020.
[2] BBC news, "Coronavirus: How does Covid-19 test-and-trace work?," BBC news website, https://www.bbc.com/news/explainers-52442754.
[3] The Japan Times, "Japan ministry issues infection codes for COVID-19 contact tracing app," The Japan Times,https://www.japantimes.co.jp/news/2020/07/03/national/japan-infection-codes-coronavirus-app/.
[4] F. KURIBAYASHI, "FUMIKO KURIBAYASHI," The Asahi Shimbun,http://www.asahi.com/ajw/articles/13565255.
[5] R. Cellan-Jones, and L. Kelion,"Coronavirus: The great contact-tracing apps mystery", BBC news, https://www.bbc.com/news/technology-53485569