Embedded Contact Tracing using Mobile Devices, Cloud, and iBeacons

Mian Hamza[1] and Jingxu Hu[1]
Integrated Microsystems Laboratory
Dept. Electrical & Computer Engineering, McGill University,
Montreal, QC, Canada

Covid-19 is a modern pandemic that has ripped through the fabric of our daily lives. We are required to limit our exposure to other people to help safeguard each other's health. Contact-tracing/early detection serves as an effective solution for the management of Covid-19 and other similar pandemic diseases. The current popular IoT contact tracing implementations rely on a mobile-centric approach where cellular phones broadcast as beacons and nearby embedded devices log beacon interaction data locally. The proposed approach is a hybrid method that utilizes decentralized iBeacons to track individuals and a centralized cloud infrastructure to communicate/store user exposure data with BLE and Cloud access. The solution addresses potential privacy concerns and offers a simple low-cost contact tracing infrastructure set-up for offices, schools, and other similar public spaces.

Index Terms—**IoT, Cloud, iBeacon, Firebase, Contact-tracing, BLE, Smart Cities**

I. Introduction

COvid-19 is an unprecedented pandemic in the modern era that has been proven difficult to manage. In light of current vaccination situations, nullifying this disease in countries across the world may take significantly more time. To help countries manage the spread of this virus, digital contact tracing offers an effective solution. The availability of low-cost embedded IoT devices and their communication protocols would enable a streamlined digital contact tracing infrastructure set up in any given location throughout the world.

Digital **Contact Tracing** comprises of using embedded IoT devices such as smartphones to detect nearby users and store them as "contacts". If an edge user i uploads a positive test result in a network of N users, all contacts who have come into close contact with user i are notified and are requested to quarantine for 2 weeks. A popular method adopted by the Canadian Government was developed by Apple-Google [1] [2]. This decentralized solution utilizes the Bluetooth Low Energy (BLE) protocol. The approach allows edge users to log interaction data with each other through BLE and ensures privacy by only sharing randomized cryptographic ids between devices. Apple-Google's **Exposure Notifications** API allowed trusted developers to realize their various digital contact tracing solutions [1].

While this framework and other derivative solutions do ensure users' anonymity by broadcasting cryptographic ids, it also presents an enormous opportunity for malicious entities to intercept and de-crypt user sensitive data [3]. Thus, in this paper, we shall examine a tracing infrastructure that eliminates the broadcasting of user cryptographic data and offers a secure cloud-based storage/communication of user iBeacon proximity and exposure data.

II. Related Work

In recent months, several contact tracing solutions have surfaced. Each solution employs a different tracing architecture using the state-of-the-art technologies [4]. In the following paragraphs, brief discussions of several similar methods will be presented and analyzed for how each addresses related privacy concerns. Then, in the subsequent section, privacy shall be discussed concerning our proposed solution.

- **Apple-Google** [1]: A collaboration between Apple and Google produced a decentralized protocol for contact tracing. This protocol was adopted and further developed by other central governments and researchers [4]. Some app/protocols such as **DP-3T** and **PEPP-PT** build on top of Apple's and Google's framework [5] [6]. The protocol uses the BLE technology to track and log encounters with other user devices. Each user's contact log is never transmitted to a central organization, instead, their auto-generated cryptographic IDs are. When a user is positively diagnosed, his/her ephemeral IDs are sent to the central authority. These IDs are generated using symmetric key protocols such as HMAC-SHA-256 and AES-128. This approach addresses user privacy concerns by the same methodology used in **BlueTrace** [7].
- **BlueTrace** [7]: Singapore's TraceTogether application utilizes this BlueTrace open-source protocol in tracing a large sum of its population [4]. It combines it with BLE technology where devices exchange anonymous auto-generated IDs through broad-cast. After a board-cast phase has been completed, the devices will log all encounters in a form of a history log. Then, when an endpoint user has experienced a positive exposure, his/her encounter logs will be sent to a central organization for inspection. BlueTrace addresses user privacy concerns by distributing short-lived auto-generated IDs to individual embedded devices through a central authority [4]. The cryptographic algorithm used to generate these anony-

Manuscript received August 25, 2021; revised January 10, 2022. Corresponding author: Mian Hamza (email:mian.hamza@mail.mcgill.ca). Both authors contributed equally to this work.

mous IDs was AES-256-GCM.

- **Hamagen** [8]: Israel's Ministry of Health developed a contact tracing application utilizing a device's Global Positioning System. Hamagen continuously tracks individuals' GPS coordinates (after initial consent has been given). This method requires no interaction with other endpoint devices. Thus, this architecture allows the identification of exposed individuals and people who have come in close contact with them through active monitoring of device GPS logs. Hamagen addresses user privacy concerns by performing consented GPS coordinates cross-referencing on the endpoint devices. Covid location data is sent to the user by the Ministry of Health and no device-sensitive GPS data is returned. In comparison with our proposed approach, iBeacons & BLE positioning provides several advantages over Hamagen's GPS based tracking. BLE background tracking applications allow for the efficient use of battery life on edge devices thus providing longer tracking periods within a network of iBeacons. It also boasts a highly accurate indoor ”Micro-location” proximity detection based on the Received Signal Strength Indication (RSSI) between the iBeacons and the edge devices [10] [11]. In an enclosed environment such as an office space or a shopping mall, iBeacons & BLE do not rely on poor satellite network signals for pinpointing edge device locations as does Hamagen. Lastly, iBeacons allow for a context-aware messaging/event-based notifying app architecture to be installed across different types of edge devices rendering it interface-able in a cross-platform setup [11].
- **Other Indoor Positioning Methods** [12] [13]: Apart from the aforementioned methods, there exists other state-of-the-art techniques for indoor tracking. They include Software-defined Radio Frequency Identification (RFID) for indoor positioning, camera-based Computer Vision & Deep Learning approaches for live time tracking, and the use of Wi-Fi networks for indoor localization. In RFID, it relies on radio antennas sending/receiving waves to identify and track near by moving objects with RFID tags acting as transponders. In a camera based approach, live feed of video frames are pre-processed & fed into an object detection and tracking inference engine utilizing a state-of-the-art CNN architecture. Lastly, in WiFi networks, connected embedded devices are localized based on their distance relative to the major access point (IAP) of the network. The tracking precision depends on the power/signal strength of the network. In comparison to our proposed approach, iBeacons & BLE was chosen for live tracking as it boasted several advantages over the methodologies described above. iBeacons can be deployed in all areas of an enclosed environment (walls or ceilings) and tracks detected BLE devices as they come & go from one beacon to another thus rendering it as device-free. This would not be the case for RFID as devices would require physical tagging as they enter a tracking area which may pose as an inconvenience for both the institution and the individual. As for the live tracking of individuals using camera modules, it requires high resolution video frames to be used for the inference engine to output correct tracked individuals and not erroneous inanimate objects. In addition the camera modules have no capability to gauge and track individual distances from each other where as iBeacons & BLE do. Lastly, embedded devices that use WiFi networks for indoor localization may find that it's battery life is significantly shortened during usage thus posing a challenge to track individuals for an extended period of time. Utilizing BLE & iBeacons would not pose such a challenge [10] [11].

III. Privacy

The proposed solution addresses privacy concerns by not utilizing the advertisement of beacons from endpoint devices. Instead, the user IDs, iBeacon UUIDs, and users' exposure statuses are only accessible through Cloud Firestore. The mobile devices will not advertise as beacons but rather set up as BLE iBeacon detection/ranging applications that communicate with a cloud server/storage platform indicating which iBeacons they have been in contact with. The iBeacon devices represent external SoCs that only advertise iBeacon information through BLE. It is possible for companies and schools to employ specific user accounts for their employees/students so that the information can only be accessed through the cloud and user app endpoints. This will in effect lock out any unwanted attempts to compromise the tracing infrastructure and the exposure statuses of its users.

IV. Proposed Implementation

The goal of this project was to develop a low-cost infrastructure to universally contact trace a moving population. iBeacon proximity data are obtained using Bluetooth Low Energy and are stored in the cloud. The system is able to determine the location of a user at a specific time by analyzing the obtained iBeacon data fields. The application allows for a positive test result to be uploaded to the database through the app. Consequently, all devices within the same location and time frame are notified of potential exposure.

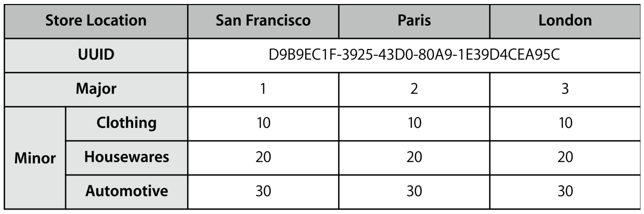
The proposed digital contact tracing solution shall be further divided into 4 subsequent sections. Each section will gradually unveil the intricacies involved in monitoring a large portion of embedded devices in a decentralized manner.

A. Technologies And Processing Endpoints

The proposed solution is implemented by integrating different software and hardware platforms together. The infrastructure is comprised of 3 major processing endpoints, namely, the **Cloud**, the **Embedded Device**, and the **iBeacon Device**. Each processing domain employs a different set of technologies.

- **Embedded Device**: The user endpoint of the tracing solution is implemented as an iOS BLE application that utilizes Apple's Core Location framework. This framework provides services that determine a device's geographic location in terms of longitude, latitude, and orientations relative to nearby iBeacons. It utilizes Wi-Fi, GPS, Magnetometer, and Barometer in order to calculate live location data. The Core Location Framework can actively perform region monitoring, beacon ranging, and live location updates [9] [11].

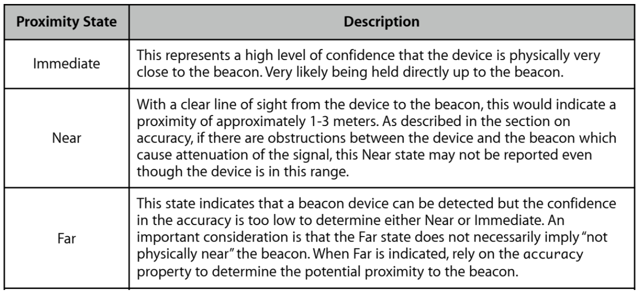
- **iBeacon Device**: The iBeacon endpoint is implemented as a self-advertising BLE process compiled onto a targeted hardware platform. The chosen SoCs included both Silicon Lab's Thunderboard BG-22 and RS9116 Wireless [15] [14]. When these beacons are activated they advertise their **Universal Unique Identifier** (UUID), **Major**, and **Minor** values through BLE to near-by peripherals. The advertised beacon data helps nearby devices range their proximity values in relation to the connected beacon within a designated beacon region. In figure 1, the mock beacon advertisement data illustrates a national retail store tracking its customers across different city locations and across different departments. The **UUID** is shared across all stores. This allows IoT devices to use a single anonymous **UUID** to recognize any stores with a single region. Then, a **Major** value is assigned to each specific region/store in order to help embedded devices determine their store locations. Finally, a **Minor** value is assigned to each specific department within a store in order to help track peripherals as they move about inside the store. Thus, with this information, embedded devices could freely identify when they have left or have entered a specific store location and what department they are currently in. The specific beacon advertisement data are determined by the host organization deploying the iBeacon technology [11].
- **The Cloud**: The cloud endpoint utilizes Google's Firebase Cloud Firestore API. This framework primarily serves two major purposes. On one hand, it provides functionalities to store a plethora of extracted iBeacon proximity/exposure data in Firestore [17]. On the other hand, it offers efficient server-side ML models such as text recognition to extract feature keywords related to an exposure event [16]. Each user i would upload their exposure results as a PNG image to Firebase's text recognizer. Then, the remote server returns a document of feature words which the app will then analyze for positive exposure. The Firestore represents a scalable database targeted at mobile, web, and server-side deployments from GCP's Firebase. This database offers real-time updates and keeps the data in-sync across all iBeacons through the use of **Listeners**. Unlike traditional SQL databases, Firestore boasts a **NoSQL**, document-oriented storage design. Tables and rows do not exist. Instead, one stores data in the form of **Documents** which are then organized into **Collections**. Furthermore, documents store data as a set of key-value pairs analogous to JSON records. Firestore is ”schema-less”, it provides the freedom over what unique key-value pairs go in each document. Lastly, either Firestore creates document IDs anonymously, or the user devices can securely generate these keys at random. [17] [18].



Store Location		San Francisco	Paris	London
UUID		D9B9EC1F-3925-43D0-80A9-1E39D4CEA95C		
Major		1	2	3
Minor	Clothing	10	10	10
	Housewares	20	20	20
	Automotive	30	30	30

Fig. 1. Sample iBeacon Infrastructure for Indoor Contact Tracing [11].

B. Simulated iBeacon Architecture



Proximity State	Description
Immediate	This represents a high level of confidence that the device is physically very close to the beacon. Very likely being held directly up to the beacon.
Near	With a clear line of sight from the device to the beacon, this would indicate a proximity of approximately 1-3 meters. As described in the section on accuracy, if there are obstructions between the device and the beacon which cause attenuation of the signal, this Near state may not be reported even though the device is in this range.
Far	This state indicates that a beacon device can be detected but the confidence in the accuracy is too low to determine either Near or Immediate. An important consideration is that the Far state does not necessarily imply“not physically near” the beacon. When Far is indicated, rely on the *accuracy* property to determine the potential proximity to the beacon.

Fig. 2. iBeacon Ranging [11].

The iBeacon architecture is based on the spacing and the layout of the location they are placed in. As shown in figure 2, this allows for the detection of 3 distinct proximity states from the iBeacon. These states include; **immediate**, **near**, and **far**. To maximize iBeacon space and accuracy, each iBeacon should be placed around 3 meters from the other. This allows for all devices within the premises to be detectable as **near** to an iBeacon at all times. Each iBeacon, in the **near** state, will cover [11]:

$$3^2 * \pi = 9\pi \approx 28.3m^2$$

In general, public health authorities recommend a social-distancing metric of at least 2 meters. However, given the scenario where people remain confined in small spaces such as offices or school classrooms for long periods, The United States Centers for Disease Control and Prevention (CDC) has stated that maintaining a distance of 2 metres (~6 feet) is insufficient in preventing a Covid-19 infection [19].

Considering the recommendations of public health authorities, the iBeacon infrastructure is set up to trace individuals who are within 6 metres of each other, either **near** or **immediate** to an iBeacon within a 5 minute time frame of each other.

The iBeacon locations are adapted to the layout of the buildings that they are placed in. For example, given a building with 7 floors with each floor spanning an area of 4000 ft^2 (372 m^2), 14 iBeacons shall be placed per level. Each iBeacon will have a **Major** value denoting the floor number. Furthermore, each iBeacon will also have a **Minor** value denoting the location of each iBeacon within that floor level. Thus, in this setup, no two iBeacons can have the same Major & Minor value combinations. This will allow us to determine the location of people relative to the iBeacon setup.

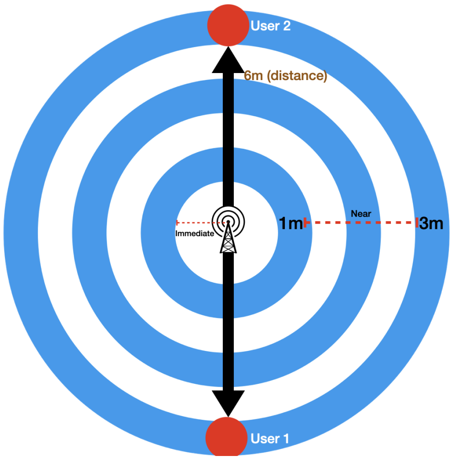


Fig. 3. iBeacon Ranging

C. Cloud Persistence

The Cloud uses services from Google's Firebase Firestore API. However, other implementations such as AWS or Microsoft Azure can be used as well. The database is set up such that each distinct iBeacon is associated with its own subcollection. Figure 4 visualizes the overall architecture of data persistence in the Firestore.

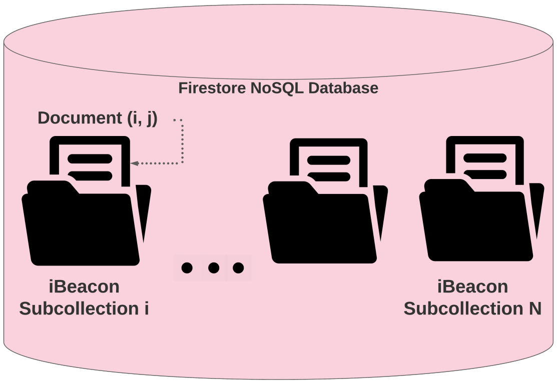


Fig. 4. Firestore NoSQL Database

When an IoT device has detected an iBeacon, it first saves it's proximity state locally. The state could either be **near** or **immediate**. Then, the device writes a struct to its subcollection comprising of 4 elements:

1) Randomly Generated Device Identifier
2) Range Value (Near or Immediate)
3) Timestamp
4) COVID Status
    - A boolean value
    - True for a positive exposure
    - False for a negative exposure

The randomly generated device identifiers keep the anonymity of the user, and since the user is writing to the cloud, no third party can determine the identity of the user. Cloud Firebase has strict rules governing its access that ensure further privacy and data security. Google regularly maintains the security and integrity of Cloud Firebase. The mobile devices will also subscribe to changes in the data-point through a **SnapShotListener** [17].

Firstly, a positive exposure result is uploaded to the cloud through the user endpoint as an image. Then, it is processed by the **textRecognizer** model from the **Firebase ML API** [16]. The model extracts feature words and detects specific bi-grams such as ”covid-19 positive” or ”POSITIVE COVID-19”.

A cloud function λ is triggered when a positive exposure result appears. When triggered, it parses through the stored documents and render the ”COVID” field value to ”True” for all users who satisfy the following 3 requirements [17] [18].

1) Share the same Major & Minor values
2) Within a 5 minute time span
3) Range is either ”Near” or ”Immediate”

After the cloud function has updated all appropriate documents according to the aforementioned update rule, the **SnapShotListener** will then notify all mobile devices that have undergone a change in it's original data values. Firestore guarantees low-latency transactions regardless of the amount of data and users. Therefore, the contact tracing system is scale-able with no degradation in its performance [17].

D. IoT Embedded Device Model

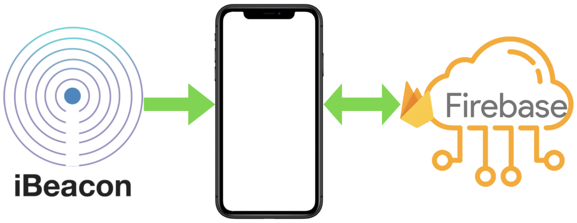


Fig. 5. Flowchart of Contact Tracing System

The user endpoint embedded device is a liaison between the Cloud and the iBeacon. It serves as a ”message” passing mechanism between the iBeacons and cloud endpoints. The architecture of the embedded application was developed for the iOS platform. Similarly, the same architecture concepts can be applied to other platforms as well. Android, for instance, could be a suitable replacement.

The user device communicates with various iBeacons in the vicinity through Apple's **Core Location Framework**. This framework contains libraries that provide functionalities such as **Location Manager** which initializes, starts, and stops the Core Location services [9] [11].

An iPhone, like many other mobile devices, can be connected to a cloud-based platform where it provides services to process data coming from various iBeacons. Firebase is a cloud platform where it allows for multi-device communication. To communicate with iOS devices, Firebase offers **Cocoa-Pod** libraries that handle inter-process communication between 2 different processing domains (Swift & Firestore).

Figure 5 depicts an overview of the contact tracing system. It begins with the mobile device detecting nearby iBeacons and then writing the beacon proximity data along with the initial exposure results to the Firestore database. Then if the user obtains a positive test result at a later time they will upload the test result through the App. The Application will, through the use of Firebase's **textRecognizer**, extract feature keywords related to a positive exposure [16]. If a positive test result is confirmed, the cloud function λ is triggered to update the Firestore database with "True" COVID boolean values for all relevant users [18]. The users whose COVID statuses have changed will then be notified by Firestore's **SnapshotListener** [17].

V. Privacy Implementation

Institutions such as provincial governments, schools, and companies can create unique accounts for their users. These accounts will then provide its users with security and cloud functionalities. These accounts serve as a medium for authentication as Firebase Cloud has security rules that grant developer/authenticated accounts to access the remote storage. No other third-party entity will be able to access the authenticated device communication channels as they are a closed-loop system where the device directly communicates with the cloud [17]. We propose a solution that can protect the identity of users from third parties and also keep a layer of privacy from institutions:

1) Authentication from central entity to write to the Cloud
2) Anonymous device IDs are generated locally from the device application and are used for Cloud functions and beacon contact tracing.

The proposed solution above prevents malicious entities from accessing and altering sensitive information regarding the users' health status. The device identifiers anonymously help contact trace which devices were present near particular iBeacons at a particular time. This two-step solution allows for not only secure cloud access but also maintains users' anonymity within institutions such as schools, offices, and governments.

VI. Experimental Results

An experiment was conducted with 20 individuals who were in close proximity to an iBeacon device that was set up inside a closed environment. The iBeacon was situated in such a position as to simulate the capture of real physical interaction trends between individuals in an enclosed space. The experiment begins by first randomly selecting an individual to submit a positive exposure result. As seen in Table I, the experiment presents the successful execution of the contact tracing steps outlined in the proposed implementation section. People who were near the infected individual in terms of time and distance were notified of a possible exposure event. This experiment simulates an infection pattern that abides by 2 specific rules; +5 mins time window and a proximity state that is either **near** or **immediate**. This meant that people who were within the +5 mins time window of the infected person and who were also close to the iBeacon device were deemed infected. The UI of the exposed users' applications is updated almost instantaneously. According to WHO, expelled Covid-19 aerosol droplets can remain airborne for a period of 8 - 14 mins from a loud conversation between a group of individuals in a small confined space. Thus, to simulate an infection pattern as close as possible to the real world, the infection time window would have to be adjusted to the values specified by WHO [20].

While the results have shown to work with a two beacon setup, the implementation is scalable and have been tested to interface with multiple beacons. In the multi-beacon setup, to determine the exact "mico-location" of the user within an enclosed environment, each beacon has their own unique major and minor values. The organization that owns the infrastructure can use specific major and minor numbers for particular locations according to their own unique numbering system [11]. However, the locations of users are not needed to conduct contact tracing. Our proposed implementation would notify users of potential exposures when they have been near the same beacon as an infected individual.

VII. Limitations

In any proposed implementation there exists limitations to how accurate or precise their measured metrics can be. In our case, edge devices have a certain range of confidence levels on the detected distance of iBeacons when they roam around it's vicinity. According to [11], edge devices utilize BLE's RSSI coming from the detected iBeacons to determine both their proximity and the accuracy of their estimation of proximity. Thus, as the devices move closer to the signal source, the confidence level increases and thus detecting a more accurate distance. For example, at a distance of 1m, the detected error can be as low as $\pm 0.1m$.

VIII. Infrastructure Costs & Benefits

While it is true that the infrastructure cost exceeds that of a simple phone setup, it is still a low cost, and superior in some aspects to using Apple-Google & BlueTrace. It is better for privacy preservation, and therefore more easily adoptable in indoor infrastructures. As iBeacons & BLE are suited for micro-location tracking according to [10] [11], this allows institutions to make quick decisions to shutdown, or in the case of offices and schools, to adopt a work-from-home environment. Given the incubation period of many diseases such as Covid-19, where the disease is not detectable but still spreads [20], it will save the organizations more time, effort, and money in the long run.

IX. Conclusion

Contact tracing is a necessary step in managing the spread of pandemic diseases in the foreseeable future. This project evaluates the use of embedded systems in contact tracing. This includes cloud, iBeacons, and mobile phones. The solution offers an effective method to build contact tracing infrastructure within settings such as schools and office buildings. The iBeacon Major and Minor values are used as location identifiers that help the system determine the location of the user at a specific time. Each iBeacon is able to cover a significant amount of area [11]. The cloud enables the system to store and process beacon proximity/exposure data. The privacy of users is preserved due to Firestore's safety rules

TABLE I
CONTACT TRACING 20 NEAR-BY DEVICES

Document ID	Device ID	iBeacon UUID	Major	Minor	Proximity	Distance (m)	Test Result	Create Time
5Acdab5DAUj3O0cafhxT	827DE77D-0D2D-4417-A2F1-73324F644BE3	E2C56DB5-DFFB-48D2-B060-D0F5A71096E0	34987	1025	2	1.291549665	FALSE	7:29:07 PM
6o3WBDHRpLCOTGPrGuhG	2E66E729-B273-4EFE-933A-9E0A28241557	E2C56DB5-DFFB-48D2-B060-D0F5A71096E0	34987	1025	2	2.15443469	FALSE	7:18:31 PM
7oWzn6bhNMyFQD4tLdN9	**4C22AB3D-24DD-480A-920F-0A2726604DBB**	**E2C56DB5-DFFB-48D2-B060-D0F5A71096E0**	**34987**	**1025**	**2**	**1.291549665**	**TRUE**	**7:10:25 PM**
7urqqS6c3mqOL04rvOTu	C64F2E4D-7227-444A-ACB2-AEB112661875	E2C56DB5-DFFB-48D2-B060-D0F5A71096E0	34987	1025	1	0.243833045	FALSE	7:23:42 PM
8ZzV1KTOXKsybT7l8jZe	D5E37A00-D456-4FA5-99A7-57F3A2B27B46	E2C56DB5-DFFB-48D2-B060-D0F5A71096E0	34987	1025	1	0.173615495	FALSE	7:17:16 PM
8ndVZCSW7ZAePF6OdQ2W	**4408B675-A659-43B3-92C6-5454C4F8796D**	**E2C56DB5-DFFB-48D2-B060-D0F5A71096E0**	**34987**	**1025**	**1**	**0.189573565**	**TRUE**	**7:08:53 PM**
DTnn9Wo3plYslZV8kmGD	FDFC3BBE-A308-47AF-8C04-A1E1AF767E42	E2C56DB5-DFFB-48D2-B060-D0F5A71096E0	34987	1025	1	0.140442426	FALSE	7:27:50 PM
Dhd75CzcdDBUg9BBiRRE	1C20053F-0ECA-403F-9383-A374FDC22268	E2C56DB5-DFFB-48D2-B060-D0F5A71096E0	34987	1025	2	2.782559402	FALSE	7:21:36 PM
GGV8knXiGVn6cc8QVKnH	**460239F2-870F-462A-9C82-D0B5AA27A871**	**E2C56DB5-DFFB-48D2-B060-D0F5A71096E0**	**34987**	**1025**	**1**	**0.188626033**	**TRUE**	**7:07:58 PM**
GysASx7FMiXu0uSECMNl	475D6BA9-4C9B-462A-977B-2322C09587ED	E2C56DB5-DFFB-48D2-B060-D0F5A71096E0	34987	1025	2	1.668100537	FALSE	7:19:52 PM
Jb5vPn7dULFx7Ym0y5fP	1F0904DD-437B-4D4C-9755-F64BBBCF1C29	E2C56DB5-DFFB-48D2-B060-D0F5A71096E0	34987	1025	2	2.448436747	FALSE	7:22:35 PM
M178kZka1QMWG3vYGEbD	5A7FA521-9BD6-48F7-B3AA-34B0A9E1FE56	E2C56DB5-DFFB-48D2-B060-D0F5A71096E0	34987	1025	1	0.59948425	FALSE	7:25:07 PM
MjuHrT2qtc6igl4WGDRJ	F261D3CA-6DC3-4621-8DC6-1A73BEC1096C	E2C56DB5-DFFB-48D2-B060-D0F5A71096E0	34987	1025	1	0.879922544	FALSE	7:18:02 PM
MstBcMGgXfQyER4GSqJN	7FBFB8BB-3DEF-45B8-9710-01029E717A69	E2C56DB5-DFFB-48D2-B060-D0F5A71096E0	34987	1025	1	0.201702482	FALSE	7:27:24 PM
Wb5cvwl498VykIozDw4E	**270A0367-6341-4F74-A19F-A911E89CE4CB**	**E2C56DB5-DFFB-48D2-B060-D0F5A71096E0**	**34987**	**1025**	**1**	**0.112343457**	**TRUE**	**7:09:47 PM**
cuviRpWxlOHojeTfHNKt	D1F36178-F176-4D2F-AB73-ED0B00C3DD07	E2C56DB5-DFFB-48D2-B060-D0F5A71096E0	34987	1025	1	0.681292069	FALSE	7:25:47 PM
pIJkHG4mvh10X6eyiVmT	D66732EF-A9A0-47EA-B3D9-F46DE4ED9426	E2C56DB5-DFFB-48D2-B060-D0F5A71096E0	34987	1025	2	2.782559402	FALSE	7:20:58 PM
qJOEka9UaoyCjXHnqeyW	C30E6285-3002-4501-A79B-B1BE6B3ABB16	E2C56DB5-DFFB-48D2-B060-D0F5A71096E0	34987	1025	2	2.130342848	FALSE	7:19:00 PM
sDUvpdqAoE1xLMCK9ZyK	D3871B11-82E2-4321-8C74-70FE91E1D249	E2C56DB5-DFFB-48D2-B060-D0F5A71096E0	34987	1025	3	3.16227766	FALSE	7:22:04 PM
sDUvfdsaAoR1xAUCK9ZuO	A1271C11-12U2-5634-8J74-70ON91U1E2I9	E2C56DB5-DFFB-48D2-B060-D0F5A71096E0	34987	1025	2	2.15627638	FALSE	7:20:10 PM

concerning access. In addition, the proposed solution offers randomly generated user IDs that only interact with Firestore. Upon uploading a positive diagnosis a chain of actions is set into motion that ultimately notifies endpoint users of positive exposure. Firestore is scalable, it enables the application to support a growing population of users [17]. To conclude, an effective yet simple contact tracing infrastructure can be implemented with iBeacons, cloud, and mobile applications.

ACKNOWLEDGMENT

1) **Kamalpreet Singh** for his earlier work on iBeacon hardware.
2) **Professor Zeljko Zilic** for his guidance.

SOURCE CODE

Project source code in the **final project** folder:

1) **https://github.com/jhu960213/ECSE-682**

REFERENCES

[1] Apple Google. (2020) Privacy-Preserving Contact Tracing. (Accessed: 2021-1-17). [Online]. Available: https://www.apple.com/covid19/contact-tracing

[2] H. Canada, "Download COVID Alert: Canada's exposure notification app," Download COVID Alert: Canada's exposure notification app - Canada.ca, 23-Mar-2021. [Online]. Available: https://www.canada.ca/en/public-health/services/diseases/coronavirus-disease-covid-19/covid-alert.html. [Accessed: 06-Apr-2021].

[3] Clearview. [Online]. Available: https://clearview.ai/. [Accessed: 10-Feb-2021].

[4] P. Tedeschi, S. Bakiras, and R. Di Pietro, IoTrace: A Flexible, Efficient, and Privacy-Preserving IoT-enabled Architecture for Contact Tracing, arXiv.org, 02-Jan-2021. [Online]. Available: https://arxiv.org/abs/2007.11928v4. [Accessed: 17-Feb-2021].

[5] Decentralized Privacy-Preserving Proximity Tracing: Overview of Data Protection and Security, https://github.com/DP-3T/documents/blob/master/DP3T%20White%20Paper.pdf, 2020, (Accessed: 2021-1-17).

[6] PEPP-PT Team. (2020) Pan-European Privacy-Preserving Proximity Tracing. (Accessed: 2021-1-17). [Online]. Available: https://www. pepp-pt.org/

[7] J.Bayetal., BlueTrace: a-privacy-preservingprotocolforcommunity-driven contact tracing across borders, Government Technology Agency-Singapore, Tech. Rep, 2020.

[8] Israeli Health Ministry. (2020) Hamagen. (Accessed: 2021-1-17). [Online]. Available: https://govextra.gov.il/ministry-of- health/hamagen-app/download-en/

[9] Developer.apple.com. 2020. Core Location [Online]. Available: https://developer.apple.com/documentation/corelocation. [Accessed: 02-Dec-2020].

[10] nishan_clrbridge, "Location-based technology for mobile apps: Beacons vs. GPS vs. WIFI," Clearbridge Mobile, 20-Oct-2016. [Online]. Available: https://clearbridgemobile.com/location-based-technology-for-mobile-apps-beacons-vs-gps-vs-wifi/. [Accessed: 30-Dec-2021].

[11] iBeacon - Apple Developer. [Online]. Available: https://developer.apple.com/ibeacon/. [Accessed: 15-Feb-2021].

[12] J-NaNA journal,Wei He, Pin-Han Ho, Dong Wang, Lizhong Xiao (2021). Efficient Beacon Deployment for Large-scale Positioning. Journal of Networking and Network Applications, Volume 1, Issue 2, pp. 40–48. https://doi.org/10.33969/J-NaNA.2021.010201.

[13] Fekher Khelifi, Abbas Bradai, Abderrahim Benslimane, Priyanka Rawat, and Mohamed Atri. 2019. A Survey of Localization Systems in Internet of Things. Mob. Netw. Appl. 24, 3 (June 2019), 761–785. DOI:https://doi.org/10.1007/s11036-018-1090-3

[14] SLTB010A EFR32BG22 Thunderboard Kit - Silicon Labs, EFR32BG22 Thunderboard Kit - Silicon Labs, 02-Feb-2021. [Online]. Available: https://www.silabs.com/development-tools/thunderboard/thunderboard-bg22-kit. [Accessed: 15-Feb-2021].

[15] RS9116 Wireless SoCs - Silicon Labs, 27-Oct-2020. [Online]. Available: https://www.silabs.com/wireless/wi-fi/rs9116-wi-fi-ncp-socs#. [Accessed: 15-Feb-2021].

[16] ML Kit for Firebase, Google. [Online]. Available: https://firebase.google.com/docs/ml-kit. [Accessed: 1-Dec-2020].

[17] Cloud Firestore — Firebase, Google. [Online]. Available: https://firebase.google.com/docs/firestore. [Accessed: 02-Dec-2020].

[18] 'Documentation; Firebase,' Google. [Online]. Available: https://firebase.google.com/docs. [Accessed: 03-Dec-2020].

[19] "Scientific Brief: SARS-CoV-2 Transmission", Centers for Disease Control and Prevention, 2021. [Online]. Available: https://www.cdc.gov/coronavirus/2019-ncov/science/science-briefs/sars-cov-2-transmission.html. [Accessed: 19- May- 2021].

[20] "Coronavirus disease (COVID-19)," World Health Organization. [Online]. Available: https://www.who.int/emergencies/diseases/novel-coronavirus-2019 [Accessed: 06-Apr-2021].

Mian Hamza (B.Eng'20) is a master's student in Electrical Engineering at McGill University. His research focuses on Machine Learning and Embedded Systems. He received his undergraduate degree in Electrical Engineering with a minor in Economics at McGill University. His current research interests also include Medical AI, IoT, and Block-chain. Hamza has worked on multiple embedded systems, and artificial intelligence projects as part of the Integrated Microsystems Laboratory.

Jingxu Hu (B.Eng'20) is a master's student in Electrical Engineering at McGill University. His research focuses on Embedded Systems Applications and Machine Learning (ML). He received his bachelor's in Electrical Engineering with a minor in Software Engineering at McGill University. His research interests include IoT applications, ML on Embedded Systems, and Hardware Verification/Security. Jingxu has worked on various hardware/software co-design projects within the AI & IoT domain as part of the Integrated Mycrosystems Laboratory.