covidQR: QR Code Contact Tracing

CS 426 (Internet of Things) Semester Project

Stephanie Greenberg
Department of Computer Science
Binghamton University
Binghamton NY USA
sgreenb8@binghamton.edu

ABSTRACT

The COVID-19 (also referred to as coronavirus) pandemic has affected the entire world and led to major changes in everyday life for many. Individuals as well as local and federal governments have put in place various safety measures, such as mask wearing, social distancing, and quarantining, to prevent the spread of the virus. While scientists and medical experts have worked hard to develop treatments and vaccines for the virus, technologists and engineers have also created new tools and programs to make important COVID-19 related data accessible to the public and to track the spread of the virus within and across communities.

Many states and countries have adopted their own form of app-based contact tracing, with some apps using GPS technology to track location and some using Bluetooth signals to track proximity between individuals. Similar to these various contact tracing applications, covidQR is an Android app that uses QR codes to track all of the public locations that a user has visited and uses that information to alert users if they may have possibly been exposed to coronavirus.

KEYWORDS

Coronavirus, COVID-19, Internet of Things, mobile application, Android, contact tracing, alert system, QR code, Thunkable

1 Introduction

COVID-19 originated in China at the end of 2019, and rapidly spread across borders and throughout the whole world. As of the time of writing this report, there have been about 63 million COVID-19 cases worldwide and 1.5 million COVID-19 related deaths. The United States was hit particularly hard by the virus, with cases continuing to spread vastly into the final months of 2020. The United States government has not implemented a widespread contract tracing program and has instead largely placed the responsibility on local governments. Some states have implemented their own contact tracing programs and apps. For example, New York and New Jersey released the COVID Alert NY app, which uses Bluetooth signals to keep track of who has been within 6 feet of each other so that if someone tests positive, they can notify all of the people who have been in close contact with that person. The reason for using this new application of Bluetooth technology rather than GPS technology is to ensure that every user's information is kept secure and that their privacy is maintained. Singapore and Australia have also implemented similar contact tracing apps that use Bluetooth technology. The use of Bluetooth signals is flawed, however, because signal strength can vary across phones and Bluetooth signals might detect a phone in a different room.

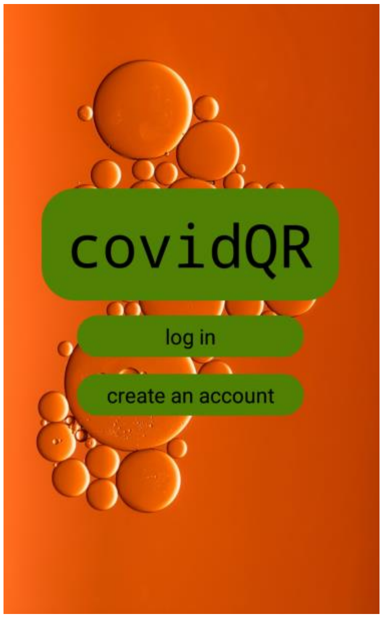


Figure 1: covidQR mobile application home page

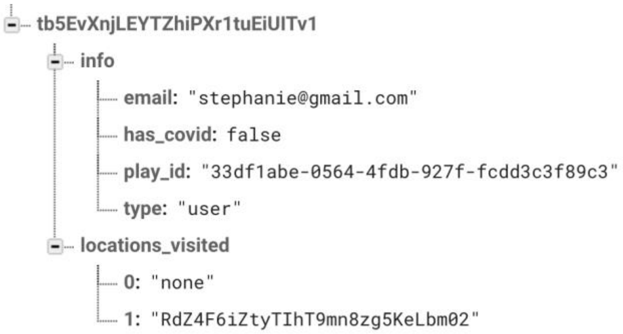
Other countries have focused less on privacy and more on ensuring that contact tracing efforts are rigorous, effective, and accurate. For example, South Korea has adopted a system that broadcasts mobile alerts providing information about where an infected person has recently been, and China uses a system that requires people to scan a QR code indicating whether or not they are at risk for COVID-19 before they enter a public place. After reading about these contact tracing efforts in China and South Korea, I was inspired to develop covidQR, which is an Android application that works by providing each user with a personalized QR code. When the user enters any public place or business such as a restaurant, movie theater, or store, their QR code will be scanned by the business's device. The app keeps track of all of the users that have visited a business, and relies on users to self-report if they test positive for COVID-19. When they do report their positive COVID-19 status, every other customer who visited the same place as them and was potentially exposed will be alerted.

2 System Design and Implementation

covidQR was designed and coded in Thunkable X and uses Firebase Authentication to store user authentication information and Firebase RealtimeDB, a NoSQL cloud database, to keep track of and update user information. The app also uses the OneSignal push notification service for sending exposure alerts to app users and the GoQR API for generating unique user QR codes.

2.1 Firebase

covidQR uses Firebase Authentication to keep track of user login information. When a user opens the app for the first time, they will navigate to the "Create Account" page and enter their email address, password, and whether they are a customer or a business. The user will then be sent a verification email; however, it is not required for a user to verify their email address to use the app. Once the user creates an account, their username and password will be then stored in Firebase Authentication so that they can sign in and access their account. Every user that creates an account is assigned a unique user ID by Firebase, and the app uses this unique ID to identify a user in the database. Each time an account is created, an entry will also be created in Firebase's RealtimeDB corresponding to that user, using their user ID as an identifier (See Figure 2). Firebase RealtimeDB stores data in JavaScript Object Notation (JSON), so I used the JSON format when adding data to the database. The user entry will contain information about the user such as their email, whether they are a customer or a business, and either the locations they have visited if they are a customer or a list of customers if they are a business. Every time a customer visits a business, the business's unique user ID will be added to the customer's list called "locations_visited", and the customer's unique user ID will be added to the business's list called "customer_list". The database also records each user's player ID, which will be further explained in Section 2.3. Each customer user also has a field called "has_covid" which is initially set to false and will be set to true if a customer indicates that they have tested positive for COVID-19. In turn, every time a customer visits a business and the business scans their QR code, it will check if the customer's "has_covid" field is true. If it is, it will send an alert on the business phone indicating that the customer should be denied entry from the establishment. Otherwise, entry will be permitted to the customer.



```
tb5EvXnjLEYTZhiPXr1tuEiUITv1
    info
        email: "stephanie@gmail.com"
        has_covid: false
        play_id: "33df1abe-0564-4fdb-927f-fcdd3c3f89c3"
        type: "user"
    locations_visited
        0: "none"
        1: "RdZ4F6iZtyTIhT9mn8zg5KeLbm02"
```

Figure 2: Firebase RealtimeDB entry for a customer app user

2.2 GoQR API

covidQR works by having businesses scan the QR code of every customer that walks in. A QR code is a type of barcode that stores different types of data, which can be read by scanning it. For my app, I used QR codes to hold text containing information about a user. Each customer has a landing page that they are taken to once they sign in, and this page displays a QR code that is uniquely generated using the GoQR API. The GoQR API has two API commands: create-qr-code and read-qr-code. I used the create-qr-code command along with parameters to specify how I wanted the QR code to look and the text to be encoded within the QR code. The data stored in the QR code is the customer's unique user ID assigned by Firebase, as mentioned in the previous section. When a business scans a customer's QR code, it will get that customer's user ID which will allow them to modify data in the database pertaining to that customer. In Figure 3, you can see the customer landing page which contains an example of a uniquely generated QR code that will get the customer's user ID when scanned.

2.3 OneSignal

OneSignal is a push notification platform that can be integrated with Thunkable X. Although OneSignal supports push notifications for both Android and iOS devices, Thunkable X does not support the ability to test push notifications on iOS. Therefore, I had to limit the use of covidQR to Android devices only. Similarly to Firebase, OneSignal assigns a unique identifier, called a Player ID, to each device that uses the app. This Player ID is slightly different to Firebase's user ID in that Firebase assigns a user ID to every user that creates an account in the app, while OneSignal only assigns a Player ID to every unique device that uses the app. As previously mentioned in Section 2.1, every time a user creates an account, their information (including their Player ID which is retrieved from OneSignal) is added to the database. By adding each user's Player ID to the database, it allows the app to associate a user's account created in Firebase with their device. This allows OneSignal to send push notifications only to specific users.

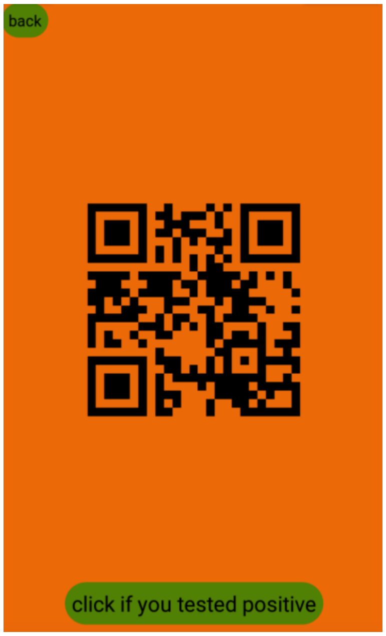
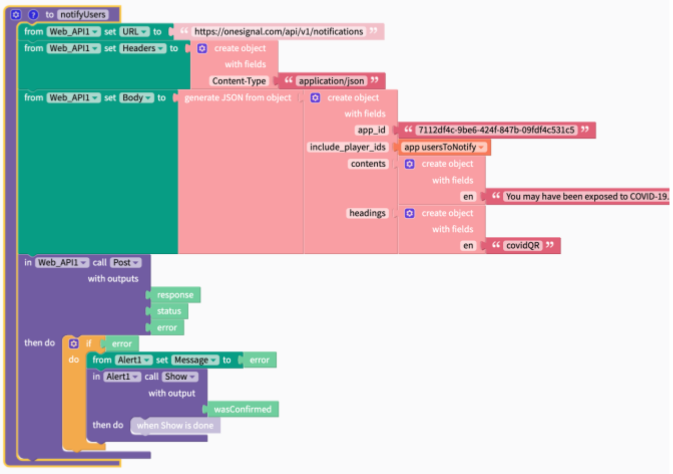


Figure 3: Customer landing page with QR code generated by GoQR API that contains unique user ID generate by Firebase. You can scan this QR code with your phone and see that it contains a user ID.

When a user presses the button to indicate that they have tested positive for COVID-19, the app locates that user in the database by their Firebase user ID and loops through all of the Firebase user IDs in the positive user's list of visited businesses, then for each of those businesses it loops through their list of customers and finally sends a notification to all of those customers alerting them that they may have been exposed to COVID-19. In other words, all customers who have been to the same businesses as someone who has tested positive will be alerted. Once this list of customers to be alerted has been established, the app sends a POST request to the OneSignal REST API. The body of the POST request message contains a list of Player IDs corresponding to devices that should be notified as well as the content of the push notification (See Figure 4).

3 Evaluation

Given that each device is assigned a unique Player ID by OneSignal, testing the app was a challenge. This app is designed to be used by a large number of people to ensure effectiveness, but I was unable to test on a large number of devices due to testing limitations. After many efforts, I was unable to use a multi-instance Android emulator on my MacBook Air, so I had to download two different Android emulators (BlueStacks and NoxAppPlayer) and use those to simulate and test customer phones. I was able to use my iPhone 11 as the business phone because the business phone does not receive push notifications, which are only compatible with Android through Thunkable X. If this app was to be used in a widespread manner, for example by a city or country, many developments and improvements would need to be made to ensure device and platform compatibility as well as scalability. However, this app provides the groundwork for a contact tracing app that uses QR codes and has potential for widespread use. The only way that any contract tracing app can be useful and effective, as many experts and governments have pointed out, is if it is widely adopted by a population (whether it be a city, state, or country).



```
to notifyUsers
  from Web_API1 set URL to "https://onesignal.com/api/v1/notifications"
  from Web_API1 set Headers to create object
                                 with fields
                                 Content-Type "application/json"
  from Web_API1 set Body to generate JSON from object  create object
                                                         with fields
                                                         app_id "7112df4c-9be6-424f-847b-09fdf4c531c5"
                                                         include_player_ids app usersToNotify
                                                         contents create object
                                                                    with fields
                                                                    en "You may have been exposed to COVID-19.
                                                         headings create object
                                                                    with fields
                                                                    en "covidQR"
  in Web_API1 call Post
               with outputs
                 response
                 status
                 error
  then do
    if error
    do  from Alert1 set Message to error
        in Alert1 call Show
                   with output
                     wasConfirmed
        then do when Show is done
```

Figure 4: Thunkable X code sample showing a POST request being made to the OneSignal REST API, including "include_player_ids" field to only notify certain users.

4 Conclusion

Many cities, states, and countries have failed to introduce widespread and effective contact tracing efforts. Using an app for contact tracing is a great way to encourage a large number of people to participate in these efforts and to get information about potential exposure to people quicker. Different countries have implemented different types of contact tracing apps, whether they use GPS technology, Bluetooth technology, or QR health codes. covidQR was designed as a contact tracing app using QR codes that can prevent people with COVID-19 from entering businesses and can also alert people who have potentially been exposed to the virus. The United States has yet to adopt a nationwide contact tracing app, even though doing so could drastically prevent the spread of COVID-19. With further development, this app design could be implemented in a widespread fashion.

REFERENCES

[1] https://medicalxpress.com/news/2020-04-contact-apps-countries.html
[2] https://www.wired.com/story/apple-google-bluetooth-contact-tracing-covid-19/
[3] https://coronavirus.health.ny.gov/covid-alert-ny
[4] https://www.vox.com/recode/2020/4/18/21224178/covid-19-tech-tracking-phones-china-singapore-taiwan-korea-google-apple-contact-tracing-digital
[5] http://goqr.me/api/doc/create-qr-code/
[6] https://documentation.onesignal.com/docs
[7] https://thunkable.com/
[8] https://firebase.google.com/