Senior Sem Expo Transcript
PATH-O-Gen Spring 2021

(Introduction)

Alonso: Hi I'm Alonso Gomez,
Andrew: I'm Andrew Mccuan,
Joshua: I'm Joshua Rodriguez,
Adolfo: I'm Adolfo Valencia,

Joshua: Welcome to Path-O-Gen, our proposed method to assist in combating the spread of COVID-19, which is still affecting the world now.

(Problem at hand)

Joshua: We chose this project because of the forefront problem facing the world today. It is Spring 2021 as of now, and we are barely coming to a slow of new COVID-19 infections with today's vaccinations becoming more and more available. Because COVID-19 spreads very easily, we wanted to assist by creating a simple to use app that everyone would be able to use passively, which will both assist themselves and others. We wanted to answer questions regarding how to stay safe meeting others, and being alerted when they may be affected by COVID-19. This would be our contact tracing. We intended to warn users when they may have been in contact with another person who may have COVID-19. With this, this should help combat the spread.

(Target Audience)

Adolfo: Our target audience is everyone. Everyone. This means from the children, to the average teenager and young adult, to the elderly. Everyone is affected by COVID-19 in some way. Some groups of people are more susceptible than others, but everyone is affected equally. COVID-19 led to a current recession in today's economy, and nearly all U.S. schools switched to alternative delivery. We all have to social distance, and many businesses are unable to operate, or are very limited in their capacity to function. Everyone was affected.

(What is Path-O-Gen)

Adolfo: Path-O-Gen is our mobile app that is designed to notify the user when they may have come into contact with someone who has COVID-19, or has a high probability of having it. Using Path-O-Gen, users can receive and send notifications when they contract COVID-19, as well as provide a hotspot map of dangerous areas where we have confirmed that COVID-19 occurred. Finally, we provide a self-examination, so that users can judge whether they should get tested and provide resources and tips for users at the touch of a button.

(Project Summary Slide)

Alonso*:* We can summarize the tools, major features and resources used as the following. Our project uses React Native to power our frontend, as well as Google Maps to provide our map to use for the user experience. We also use Node.js for our REST server, which allows our frontend to easily interact with it. We used Heroku to host our REST API so that the API can interact with our intended devices. For our database, we use Postgres, and for our Backend Algorithm, we've taken use of C++ to handle our backend calculations. Both of which are on a DigitalOcean server, using crontab and supervisor to handle uptime. Together, with the Node Server, this forms our backend and allows our app to function as we need. Finally, and most importantly, we use Bluetooth Low Energy. Bluetooth Low Energy allows us to use the passive nature of it to create the interaction we need between users, while also keeping battery consumption lower than traditional Bluetooth, also known as Bluetooth Classic.

(Project Architecture Slide)

Joshua: Our Project Architecture can be summarized by the following Data flow diagram. We implemented our project as the following: First and foremost, we have two users using our app. Using Bluetooth Low Energy, they communicate with each other, sending an identification for that communication instance to each other. Once this occurs, this is sent to our backend server. For our backend server, we are using our REST API to handle this data. This REST API will respond to requests from the app, such as login requests, hotspot map information sending, and contact information sending. Using these requests, the REST API will directly interact with the database to perform CRUD operations to satisfy the request sent. From the SQL server, which was created in Postgres and hosted on a virtual machine on a DigitalOcean Droplet. This is what stores our data, 24/7. Finally, our tracing algorithm is a C++ program that is running on a different Digital Ocean Droplet that looks for notifications from the Postgres server, and then acts upon receiving that notification. In motion, our architecture is a chain reaction of requests from our frontend, to the SQL server, which is processed by our Tracing Algorithm.

(Implementation Methodology Slide)

Andrew: Our implementation timeline and project management workflow followed an Agile methodology. Regularly, our team would meet up and discuss what is most important to work on for the following weeks ahead. We were able to restructure priorities and modify existing pieces of our project structure as we were working and iterating through our app development.

(Implementation Timeline Slide)

Andrew: Here is our final timeline of when things were completed. Originally, during our Fall 2020 semester, we planned to have our backend finished before the Spring 2021 semester started, then begin the frontend implementation. However, as the semester went on, our requirements as we built the frontend shifted as we moved forward. This resulted in finishing the backend concurrently with the frontend, as well as shifting other goals around.

(Demo: Backend)

Joshua: First of all, we'll start with what the app will interact with to get its data, as this is not explicitly evident from the user frontend side. I was in charge of implementing and creating this interaction. Our App will make HTTP requests to our backend Node.js REST API. These HTTP requests include GET, and POST requests, sending and retrieving data from the server. Once a request has been received by the server, it will make SQL calls to the Postgres Database to modify, and/or receive the data. Because our backend server is an open web-facing API, I have created verification routes that are called upon each HTTP request. These verification routes check if the user was authorized, and authenticated, then if they are, the HTTP requests from the frontend will follow-through. This will also ensure users can only ever get data that relates to them, as no user will be able to call the API using their information to get information from another. Finally, as you can see here, this is an E-R diagram of our database, which displays how our database is structured. This will give context to what is exactly stored in our database, which includes entities such as our Users, and their related information such as who they contacted with, what notifications they received in the past, when they were infected and the locations they have visited shortly after contact with another app user.

(Demo: Algorithm)

Alonso: To alert users for when they have been near someone who has been infected with COVID-19, I created an algorithm that is capable of pulling tremendous amounts of data and compute this data to determine the likelihood of a user being infected with COVID-19 and advising them to social distance and to take a COVID-19 test. Our algorithm works by receiving a notification from our database once someone has been inserted on our infected table. That users ID is passed in through the notification and the algorithm is then able to gather all the potential contact user IDs and all of the corresponding location coordinates between the infected user and the potential contacts.The coordinates are then compared and an average distance between the two users for a 15 minute duration is used to determine the potential contacted users threat level. The closer the distance the higher the threat level. This threat level is then sent out as a notification to the respective users phones. After the location coordinates have been compared they are then deleted from the database as they are no longer necessary. Any non used data is deleted after a 2 week duration.

(Demo: IRL Bluetooth):

Adolfo: Here, we will be demoing our app in action. I was in charge of researching and implementing our Bluetooth functionality into our React Native app. Our app is heavily affected by the bluetooth detection of phones to each other, as this is what allows contact tracing. So first and foremost, we'll launch our two phone's apps using existing user's on our login credential page.

Once launched, Bluetooth starts scanning for other user's and advertising the current user. As you can see according to the console output, we can see the user's logging in, and outputting that device's Bluetooth ID for demonstration purposes. After waiting for a few moments, the phones detect each other. The console log here outputs the other user it sees, as well as when they see it, which is imperative to our data flow. The two phones will then send the Bluetooth ID that it had seen, which is stored in the User database table upon login, to find who it saw, then store the meeting of the two people in the database. This is how we use Bluetooth Low Energy to power one of our project's main features.

(Demo: Front-end)

Andrew: On the launch of the app you will be prompted to a login screen where a user will be able to sign into their account. If they do not have one, they can create a new one in the register screen as shown here.

In both of these screens the app will send a fetch call to the REST API with the users email, and password, then retrieve a custom token on success the user will be logged in the app or failure then the user will be notified they entered the wrong information.

Once logged in the user will be sent to the 'User Info' screen and can see information about their account such as a users threat level from potential contacts, green being safe and red being unsafe. At the bottom of this screen is a logout button which the user can use to logout of their account.

The users info screen also has a button to notify others that they have been near someone who has contracted COVID-19, on clicking this it will take you to another screen where a user will be prompted to insert a password before then can send a notification to others that they have been in contact to warn them.

This will send a post notification using FCM (Firebase Cloud Messaging) to the user's phones and update our database. FCM will handle the android based notifications. This means our user account who was contacted will get a notification that they've been near someone with COVID-19.

Here's the notification, and you can hit "OK" to dismiss it.

With this, they can also see their past notifications from the bell icon in the top right of the ‘User Info’ screen which allows users to see their notifications with the time since they received the notification. These notifications are stored on the database where they are being grabbed in JSON format with a time and it is then listed on the screen.

Adolfo: In the ‘Resources and Tips’ screen users will be able to see recommended tips on how to stay safe from COVID-19. The user will be able to view multiple tips on how to stay safe while going out to public areas. Each tip will have a description saying how it works and when it's effective to follow it.

Andrew: In the ‘Self Evaluation’ screen the user will have the option to start a self evaluation test of symptoms for COVID-19. Here they can see there is a disclaimer at the bottom of the screen informing users that this test can not fully assess if they are sick. Once in the test the user will be able to see all the questions with true or false statements, each of these answers are weighted differently by the CDC recommendation, such as “Shortness of breath or difficulty breathing” is one severist symptom of COVID-19. After completion of the self evaluation the user will be prompted with a recommendation of what to do, they can also go back to the evaluation start screen and retake the exam again.

Now we will be switching to the ‘Hotspot’ screen, here you will see a Google Maps screen with the location of the user. In the maps are markers within 50 miles of the user that show locations that have had recent contacts where users got infected. These points are being fetched from the database by using the REST API where the user's current location is sent to the database and then the device will be sent back a JSON list of (latitudes and longitudes). After this we map out all current points in the list to the map and give the user info on more unsafe areas to visit. These points are shown as translucent red circles that cover an area to give an estimated unsafe region. When a user zooms out from a circle a marker will be placed to help the users find these areas, and with more zooming out the markers will cluster and have a number of how many regions are in a specific area. The user can then click on these clusters and markers to zoom in on the hotspot region.

(Project Learning Goals)

Joshua: I had the task of creating, and managing our backend. This included our database, our server as well as the locations to host such applications. While I stayed with the familiar choice of mine of Postgres as our database, there were still various features of Postgres that I did not know beforehand. As well, I learned a lot between backend server REST interactions between a SQL database and a frontend mobile app.

Andrew: I was in charge of creating the frontend of our project, in doing so I learned about building a mobile app with React Native and Javascript. Also, in creating this project I learned how to hook into a custom built REST API and pass data between the mobile device and the

database. And, finally I learned how to build a custom hotspot feature for google maps to show locations of unsafe areas that had recent COVID-19 cases.

Alonso: I created our algorithm to collect and compare our location data. I learned to connect with postgres via C++. As well as trying to create efficient data queries as to only pull data that was necessary to keep our processing time as quick as possible. While interacting with the database, I saw various improvements that could be made and collaborated with our group to modify the database to better suit our frontend and backend needs.

Adolfo: I helped with the frontend and bluetooth communication. With these tasks I learned javascript, java, and react native. In this I also learned how to bridge between android studio java code and react native javascript code. I was able to send data back and forth between the two. Because Bluetooth was such a core feature of our project, this was imperative to our success.

(Future Plans for Path-o-gen)

Alonso: There are various improvements that can be made to our project. We can optimize our tracing algorithm to be more efficient regarding locations, such as comparing locations when there are many users interacting at the same time in the same place. As well as, the backend security and data integrity could be improved as we deal with sensitive information from the user frontend. Moreso on the backend, we could do hotspot map calculations beforehand to aggregate points to provide a cleaner hotspot map for the user experience. Frontend side, we could expand the user profile page to more accurately advise them with personalization options. Finally, while we only built for android due to the native code required for Bluetooth, we would love to incorporate iOS compatibility as it still has a sizable user base.

(Thank you)

Joshua: As for any questions or comments about our project, these can be forwarded to the following email addresses. We are glad to hear that the vaccine for COVID-19 is being administered more and more every day, which is cutting down the rate of COVID-19, and while this lowers our app's potential effectiveness in combating spread, we feel that this project and idea can be used for other fast spreading illnesses that plague us today. Thank you.