**Bold Red indicates next slide**

**Gabriel**

1. Hello we are team Last but not Least and this is our project COVIDSCPAE with Adi, Myself (Gabriel), Shou, and **Wanwiset**

2. Because of the COVID pandemic, we take the risk of exposing ourself to the virus when we go out in public and are around other people, but how much exactly is the risk? Someone may be Christmas shopping or go to an outdoor dining area, and they may deciding between different locations, and COVIDSCAPE is here to help provide an idea of what the risk may **be**

3. With geographical risk score and personal health safety measures, such as masks and social distancing, a risk score may be determined based on official data and medical studies. Also it can back track when a user reports to be diagnosed with the virus that will notify possible users that could have been in **exposed**

4. With the functionality and features of COVIDSCAPE, it is designed with a user friendly interface on both desktop and mobile devices. The risk score prediction subsumes two types of risks, the location risk score generated by the machine learning algorithm that predicts COVID severity per area up to 3 days, and static risks that act as multipliers to the location risk score. And the notifications for possible exposure backtracked to a diagnosed user will contact all possible users through text and **email**

**Shuo**

1. Hi everyone I’m Shuo and I will go over the overall architecture of Covidscape

2. First the **Client** is a web application used to interact with users.

3. The client receives various **users inputs** as shown here for different user requests.

4. The client also receives input from the **GPS Sensor** for current location.

5. Another consumer of GPS sensor data is **Node-RED** flow. The Node-RED flow achieves an automated data collection and stores data into **S3** bucket

6. User requests are now formulated by Client and sent to **Server**.

7. Covidscape also has an **Analytic Node** which utilizes PySpark. This node takes daily testing results to train models and predict risk scores. Results are also saved into S3.

8. The **Server** reads the predicted scores and user GPS data from S3 bucket.

9. Finally Covidscape uses **AWS for notification**. By using the simple notification service and simple email service, Covidscape can notify users via email and text.

10. These are the main components of the Covidscape architecture.

**NEXT SLIDE**

11. Now I will talk about our server implementation first.

**NEXT SLIDE**

12. Covidscape’s Server is implemented in Python using Flask and Boto3 to communicate with Client’s Web Application and interact with AWS

13. Flask is a popular open source micro web framework in Python providing a simple interface for the Server to handle different Client HTTP requests.

14. Boto3 is Amazon’s SDK for Python which provides simple API calls to read files from s3 bucket and send notifications using SNS and SES.

15. The client sends three main types of requests.

a. First is a request to register a user where the server stores user information and performs email verification.

b. Second is a request for risk score where the client provides the required user inputs and server responds with the predicted risk score.

c. Third, the client can report that a user has been diagnosed with COVID-19. This triggers a contact tracing on the diagnosed user and notifies those that may have been in contact.

**NEXT SLIDE**

16. The Server’s Risk score calculation consists of the location-based score as the base score and static risk factors as multipliers.

17. The Location-based Risk Score is predicted by the Analytic Node daily and saved in S3. We will discuss this in more detail later.

18. The Server finds the nearest city or region to the user’s GPS location. This is achieved with a linear search for the closest distance given by Haversine formula which gives the distance between two points. The server then extracts the predicted risk score for the closest region for the specified day.

19. Next the server multiplies the base score by static risk multipliers. Our prototype design has three of them – social distancing, wearing masks, and indoors vs outdoors. Each is assigned a different value based on medical studies.

20. Finally the score is normalized between 0 and 1 and an experimental threshold of 0.2 is chosen to indicate whether the user is in a relatively higher or lower risk.

**NEXT SLIDE**

21. Now, whenever a user reports a positive test result, the server performs contact tracing

22. A history map of users’ locations is built from the GPS data collected by Node-RED. Then the server compares the diagnosed user’s GPS location and timestamps within preset tolerances.

23. Those users who are in potential contact are notified by email and text. On the top right are screenshot examples of the notifications.

**NEXT SLIDE**

24. Apple and Google offer a similar functionality called Exposure Notifications. Covidscape’s approach is different in several ways.

25. Covidscape does not require Bluetooth functionality. This avoids missing devices that fail to establish Bluetooth connection for various reasons.

26. Next, Covidscape takes a more centralized approach and keeps the storage and computation on the cloud as opposed to Apple and Google’s approach which adds workload to device cpu, storage, and battery drain.

27. And Covidscape is not dependent on the user device’s OS version or specific hardware other than requiring internet connection and GPS sensor.

**NEXT SLIDE**

**Wanwiset - Presentation**

1. Hi everyone, I am Wanwiset and I will be showing you guys the demo of our COVIDSCAPE application.
2. So, first of all, we implemented our project as a web application. And as you can see, this supports both desktop and mobile versions.
3. This is the home page and the main functionality. The first thing you see is this map and some circles. And this extends to the whole LA area.
4. The circle's size represents COVID concentration in the area. Larger circle means there is a higher density of COVID infected population.
5. The circles are also color coded by COVID growth rate. Red means increasing and Green decreasing. As you can see, there is currently no green circle in LA.
6. To the top left and top right is our localization functionality. Use it to select the interested location. Let’s say I want to visit Hollywood.
7. Now let’s see below the map, you can choose the options to best reflect the COVID precautions at that location.
8. You can also choose to predict risk scores in the future, if you don’t want to visit the place right now.
9. Once predicted, the server returns a final score.
10. Let’s try switching out the mask and see how it affects the risk score. You can see that the risk increased.
11. The next page I want to show is the Report page. Users who are diagnosed with COVID can press this button to notify others who were in close contact with them to be careful.
12. Click the button, enter your username and I’ll receive an SMS and email notification. Here is the email, just recently received.

**NEXT SLIDE**

1. Now, I’m going to show you how we arrive at the prediction score. This functionality is handled by the Machine Learning Analytic node
2. First, the dataset is provided by the LA County Department of Public Health. They release a daily article of total COVID cases per sub-county area.
3. The data is also given as a normalized rate of cases per population. This way we can compare cities of different sizes.
4. To better represent the currently infected population. We choose a 14 days concurrent window of this data as the input.

**NEXT SLIDE**

1. For the Machine Learning algorithm, we selected to use a linear regression model. This best fits this dataset because we are predicting a future trend from historical data.
2. Let's take an example of Expo Park data points. It shows a clear upward trend and there is not much fluctuation of the values. Then, the algorithm extrapolates the red linear curve to predict future values.

**NEXT SLIDE**

1. Another consideration we made is how many historical data points to use. If we consider too long of a history, the model would not represent recent values. Here, in using long history, the cases have been rising but the curve fails to predict this.
2. To compare, we evaluate the accuracy of our predictions, which is calculated by taking the difference of the predicted value and the actual value that is later provided by LA County of Public Health
3. In this table, the accuracy of 14 days long history and 7 days short history is compared. On average, the short history performs 1.2% better with a total of 93% accuracy.

**Wanwiset - Video**

1. Hi everyone, I am Wanwiset and I will be showing you guys the demo of our COVIDSCAPE application.

2. So, first of all, we implemented our project as a web application. And as you can see, this supports both desktop and mobile versions with this sliding menu.

3. This is the home page and the main functionality. The first thing you see is this map and some circles. And this extends to the whole LA area.

4. The circle's size represents COVID concentration in the area. Larger circle means there is a higher density of COVID infected population.

5. The circles are also color coded by COVID growth rate. Red means that there is more than 25% increase in covid cases in the past week and green is the opposite. As you can see, there is no green circle in LA currently

6. To the top left and top right of the map is our localization functionality. It should be used to select the interested location before predicting the score.

7. Let’s say I want to select my location or say, I want to visit Hollywood. It’s going to center the map and show a red pin.

8. Now let’s see below the map, there are some options to select when predicting the risk.

9. You can choose the options to best reflect the COVID precautions at that location

10. Sometimes, you might not want to visit the place right now. You can also choose to predict risk scores in the future, up to 3 days ahead by this ratio button.

11. Once predicted, the client makes a HTTP GET request to the server who then returns a final score.

12. Let’s try switching the mask off and see how it affects the risk score. As you can see the risk increased. And now try some future values.

13. The next page I want to show is the Report page. Users who are diagnosed with COVID can press this button to notify others who were in close contact with them to be careful.

14. Click the button, enter your username and I’m receiving notifications in my email and phone. I will show you my email, here is a recently received notification.

**NEXT SLIDE**

15. Now, I’m going to show you how we arrive at the prediction score

16. This functionality is handled by the Machine Learning Analytics node

17. First, the dataset is provided by LA County’s Department of Public Health. They have a daily article of total COVID cases per city, each city is a sub-county area.

18. Another representation of the data is given as a rate of normalized cases per 100,000 population. This way we can compare cities of different sizes.

19. To better represent the currently infected population. We choose a 14 days concurrent window of this data as the input.

**NEXT SLIDE**

20. We used PySpark as our Machine Learning framework due to its map-reduce functionality. It will be useful for analysing aggregate data such as the overall average or getting the top ten hotspots.

21. For the Machine Learning algorithm, we selected to use a linear regression model. This best fits this dataset because we are predicting a future trend from historical data.

22. Let’s take a look at an example of Expo Park data points. It shows a clear upward trend and there is not much fluctuation of the values. The algorithm calculates the best fit line as shown in red. Then, it extrapolates this curve to predict future values.

**NEXT SLIDE**

23. Another consideration we made is how many historical data points to use. If we consider too long of a history, as shown on the left, the model wouldn’t represent recent values. Here, in using long history, the cases have been rising but the curve fails to predict this.

24. Finally, we evaluate the accuracy of our predictions. Accuracy is calculated by taking the difference of the predicted value and the actual value that is later provided by LA’s County of Public Health

25. Here, we observe that the prediction gets less accurate as we predict further from the current day. This accuracy drop is the reason why we choose to limit the predictions to 3 days in the future.

26. Then, the accuracy of 14 days long history and 7 days short history is compared. On average, the short history performs 1.2% better with a total of 93% accuracy. The reason we choose data points for 7 and 14 days is to avoid effects of specific weekdays or weekends on the test numbers.

27. Finally, the score is normalized from 0 to 1 and uploaded to the server.

28. We also map the scores to the coordinates of the city to be represented in the map of the homepage. For a total of 232 cities or sub-county areas.

29. This process is repeated and updated daily.

**Aadi**

Slide 1/25

This is a skeleton of my implementation. Firstly, OwnTracks extracts GPS co-ordinates from user’s phone and sends the payload to a bucket hosted on WebHook relay.

Next, the Webhook relay agent uses its internal network to relay data to node-red which is being hosted on ec2 instance This node-red instance saves the file locally and pushes that file to an S3 bucket, where the file is used for ML implementation.

Slide 2/26

We are using OwnTracks to get GPS data from users. Now the question is why OwnTracks?

The simple answer would be, because it’s user friendly, and avoids the hassle of running a separate service to communicate with the bucket. Additionally, it encrypts the payload it generates.

Slide 3/27

The data being sent from OwnTracks, looks like this. It has latitude, longitude, userID and timestamp. Those are the only necessary components for our project.

Slide 4/28

WebHook Relay is a relay service which uses its internal network to relay incoming traffic to a target destination. As a result, no public IP is involved.

In addition to that, it has its own flow in Node-RED, which connects it directly to the WebHook bucket.

Slide 5/29

Now coming to EC2. We used it as a centralized node for hosting node-red. This approach was convenient for both users and developer perspective. It has readily accessible resources which make it an ideal platform to host node-red.

Furthermore, EC2 can be configured to restrict certain or all incoming traffic for added layer of security.

Slide 6/30

Now the implementation of the flow.

Firstly, the data gets pushed into the node-red using this webhook node. It is configured and connected to the bucket we created in webhook relay.

Next, the function node returns the body extracted from the payload. Now, the OwnTracks node decrypts the data that was originally encrypted and returns it in JSON format. The function node connected next returns only necessary bits from the payload such as latitude and longitude. These bits are comma separated and are stored in a csv file.

Finally the data is pushed to S3 bucket using this S3 node.

**Gabriel**

Here we have some possible future work enhancement

In the user level we can broaden the range of the COVIDSCAPE beyond the LA county

On a more technical level we can add server redundancy to have multiple servers tend to multiple **clients**

Now this concludes our presentation, thank you very much

Conclusion

COVIDSCAPE aims to provide a personalized risk prediction so that users can make an informed decision when they must go outside