COMP590 REPORT

1. Title

Epidemic real-time data query mobile application (Android)

Application Name: Covid19

2. Team Members

Research Instructor: Ray Simar

Ansar Ainiwaer (aa171)

Weihong Xie (wx19)

Zhiyi Huang (zh36)

Wanyu Zeng (wz42)

Han Jiang (hj32)

Chien-Yeh Lin (cl128)

Nianyi Wang (nw19)

3. Abstract

As the world enters the post-epidemic era, the emphasis on epidemic data is waning. And now, people have gradually weakened their concern about the epidemic, but this does not mean that the epidemic has gone away from us. And as people learn more about the virus and have more data on the outbreak, we have more metrics to measure the epidemic's impact on us. The APP developed by our team provides users with multi-directional data to better understand the epidemic situation in the United States and even better see the risk factor of the epidemic situation in their state. We used Covid Act Now to provide us with data API. And

since the App is more geared toward Rice students, we have added a link to the school's appointment for a COVID-19 test.

4. Introduction

The Covid App is an application that focuses on Rice University teachers and students to better access real-time data on Covid. And in order to facilitate the teachers, students, and staff of the whole school to have a better experience, an interface for booking a new crown test and an interface for the school's medical emergency have been added. The entire program is completed on Android studio, built with Gradle, mainly using Java and XML to complete. Use Figma at the front-end level to design.

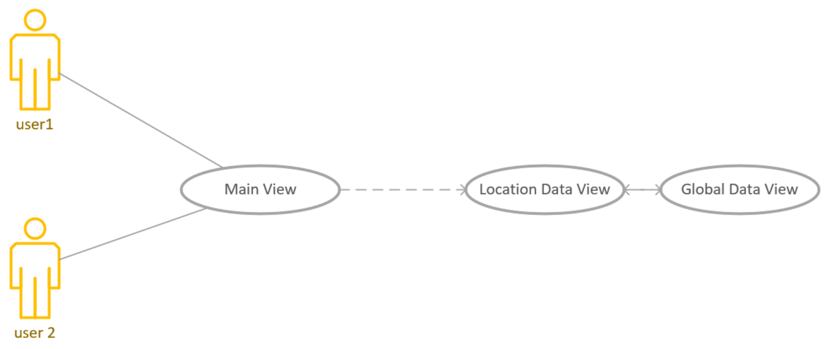
And the API interface of Covid Act Now is used to obtain epidemic data. The entire application is published and maintained by Firebase.

Android version used during development: 10.0.

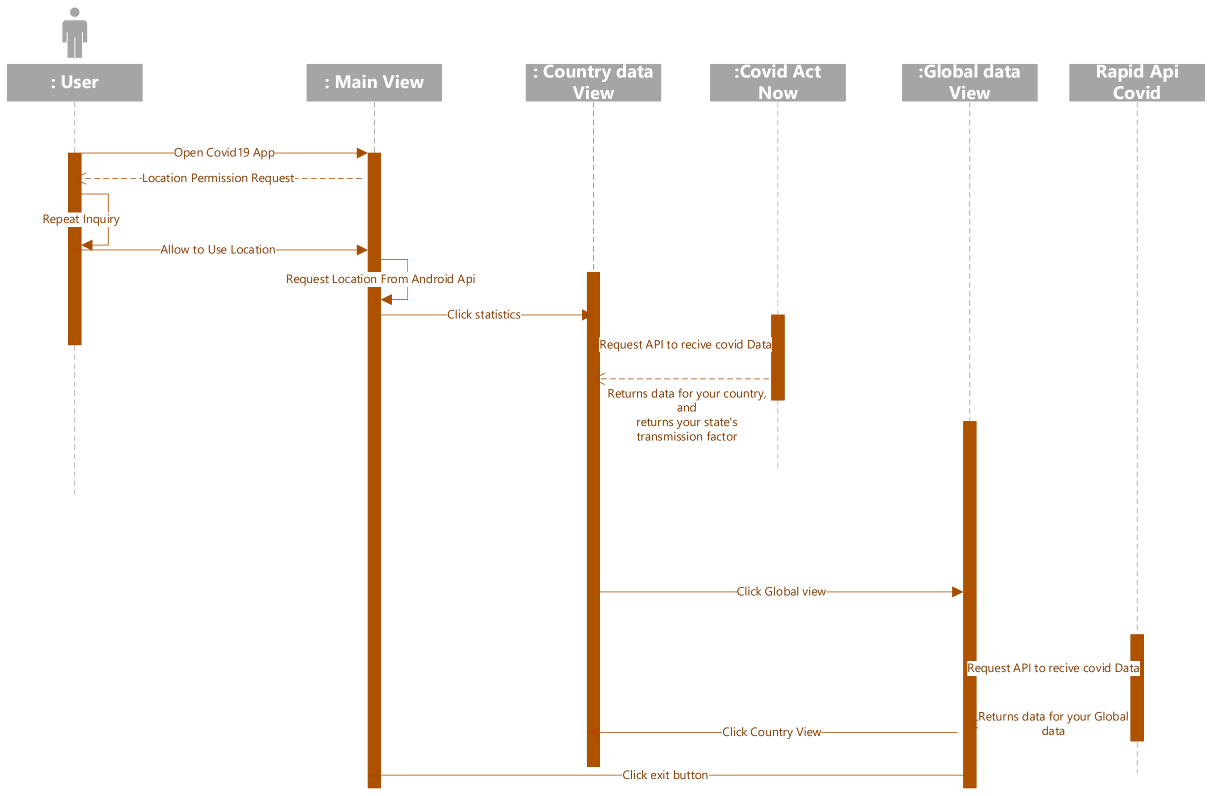
Model used during development: Pixel 4 API 29

5. Details

5.1 UML Use Case Diagram



5.2 UML Sequence Diagram



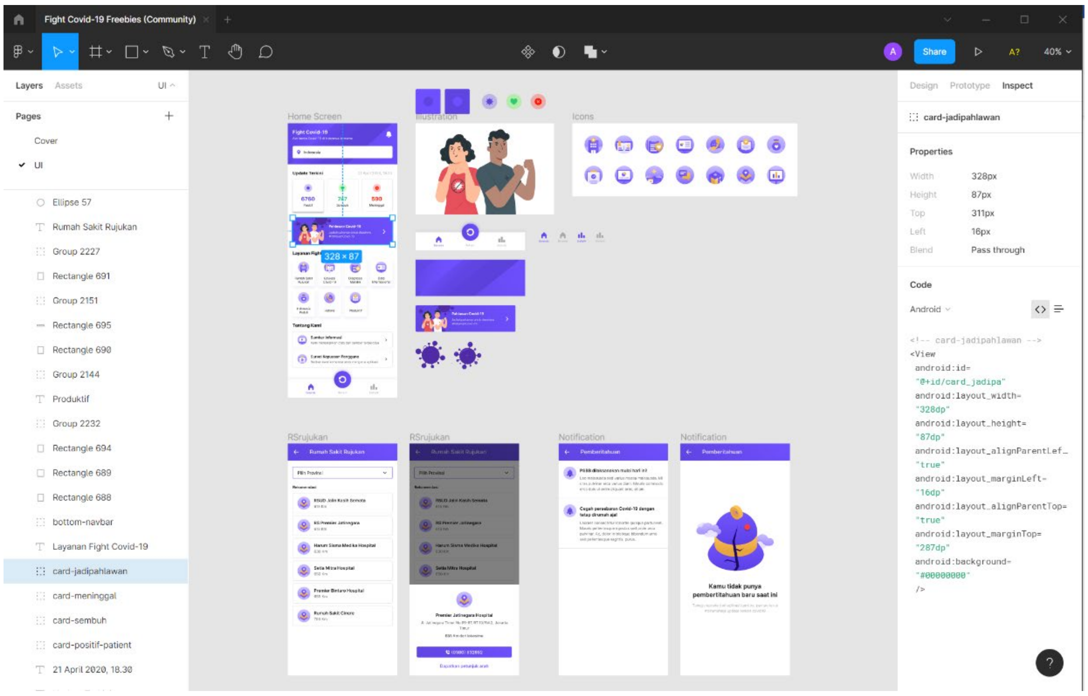
: User
: Main View
: Country data View
:Covid Act Now
:Global data View
Rapid Api Covid
Open Covid19 App
Location Permission Request
Repeat Inquiry
Allow to Use Location
Request Location From Android Api
Click statistics
Request API to recive covid Data
Returns data for your country, and returns your state's transmission factor
Click Global view
Request API to recive covid Data
Returns data for your Global data
Click Country View
Click exit button

5.3 Development

5.3.1 Front-End

In the front-end development process, we used Figma, which is more popular at this stage, for design. We can realize our front-end design through no-code operations, and the Figma community has a wealth of front-end materials for us to expand. Therefore, Figma saves us a lot of time during front-end development.

We have a total of 5 large interfaces in the whole front-end, namely activity_bot.xml, activity_global_statistic.xml, activity_main.xml, activity_splash_screen.xml, activity_statistics.xml, and other components used for front-end design.



card-jadipahlawan

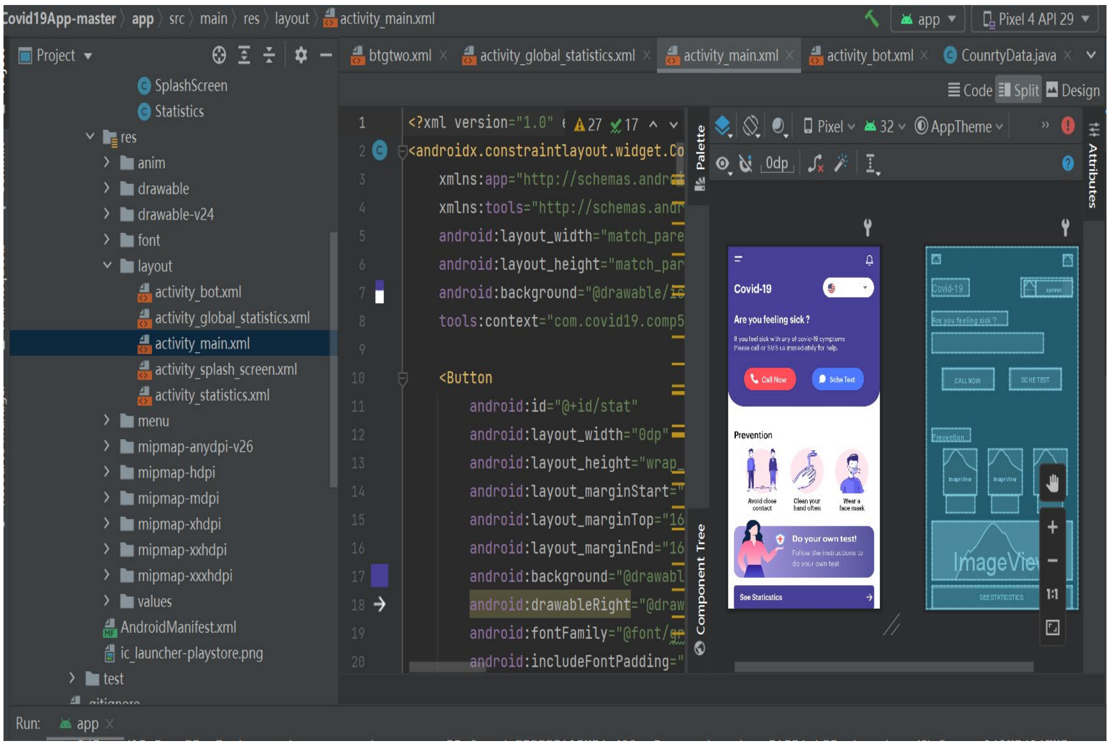
Properties

Width	328px
Height	87px
Top	311px
Left	16px
Blend	Pass through

Code

Android

```
<!-- card-jadipahlawan -->
<View
 android:id=
 "@+id/card_jadipa"
 android:layout_width=
 "328dp"
 android:layout_height=
 "87dp"
 android:layout_alignParentLef…
 "true"
 android:layout_marginLeft=
 "16dp"
 android:layout_alignParentTop=
 "true"
 android:layout_marginTop=
 "287dp"
 android:background=
 "#00000000"
 />
```



```
<?xml version="1.0" e
<androidx.constraintlayout.widget.Co
    xmlns:app="http://schemas.andr
    xmlns:tools="http://schemas.andr
    android:layout_width="match_pare
    android:layout_height="match_par
    android:background="@drawable/ic
    tools:context="com.covid19.comp5

    <Button
        android:id="@+id/stat"
        android:layout_width="0dp"
        android:layout_height="wrap_
        android:layout_marginStart="
        android:layout_marginTop="16
        android:layout_marginEnd="16
        android:background="@drawabl
        android:drawableRight="@draw
        android:fontFamily="@font/gr
        android:includeFontPadding="
```

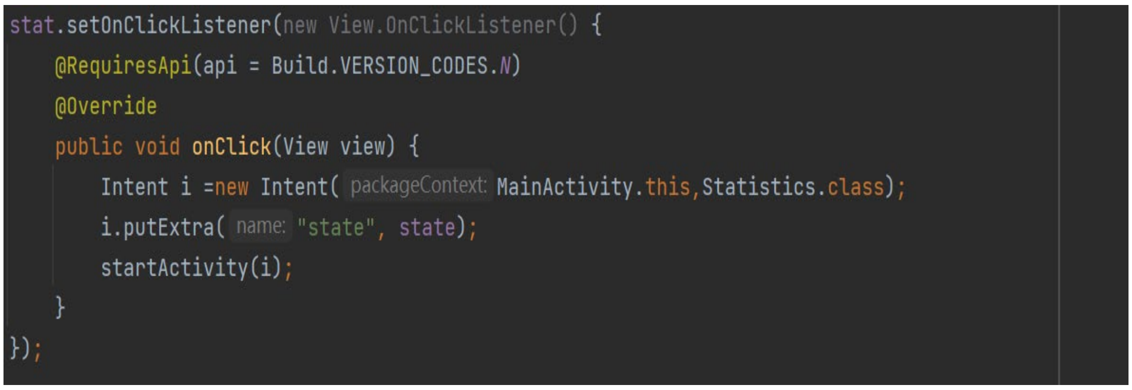
5.3.2 Back-End

In the back-end development process, we extend AppCompatActivity for the back-end development of each interface. They are bot.java, CountryData.java, GlobalStatistics.java, MainActivity.java, SplashScreen.java, Statistics.java. CountryData.java and Bot.java are components used for back-end development, so they will not be introduced in detail. And SplashScreen.java is also a loading page, so that I won't go into details here.

MainActivity.java:

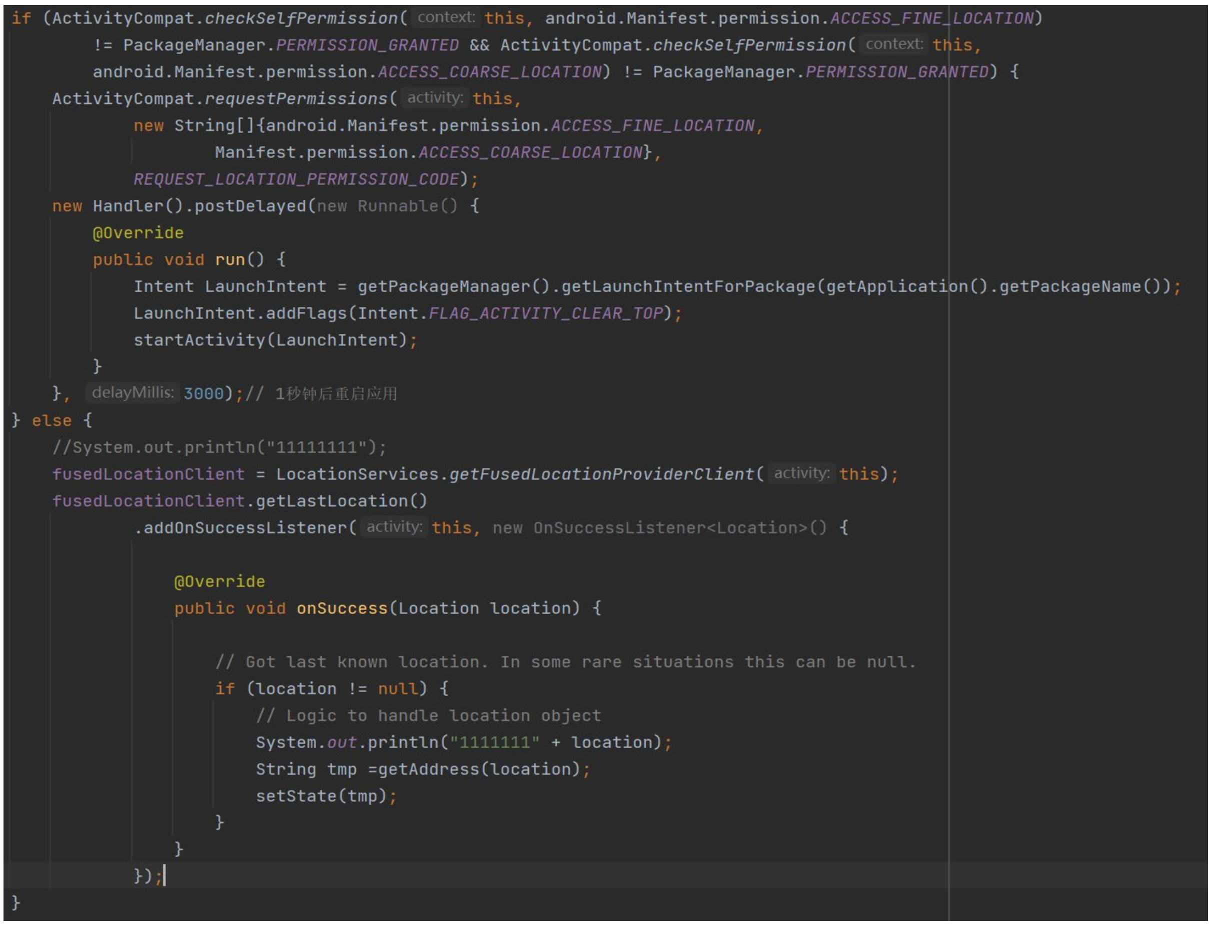
In the back-end of the main interface of the Covid19 APP, what we need to do in this class is to obtain the location permission and get the user's location through the API that comes with Android, and switch the interface through the Intent, when the user clicks the statistics button to switch to the Statistics interface.

Intent code:

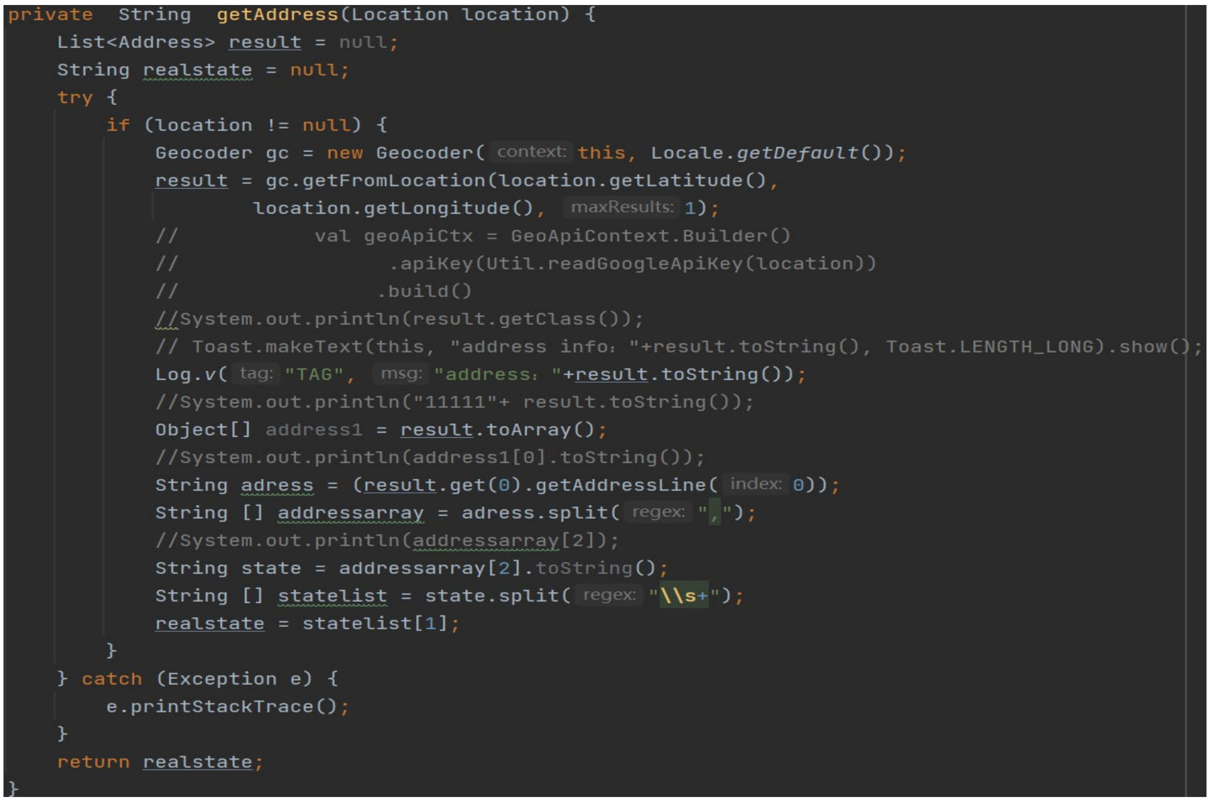


```
stat.setOnClickListener(new View.OnClickListener() {
    @RequiresApi(api = Build.VERSION_CODES.N)
    @Override
    public void onClick(View view) {
        Intent i =new Intent( packageContext: MainActivity.this,Statistics.class);
        i.putExtra( name: "state", state);
        startActivity(i);
    }
});
```

Obtain the location permission:



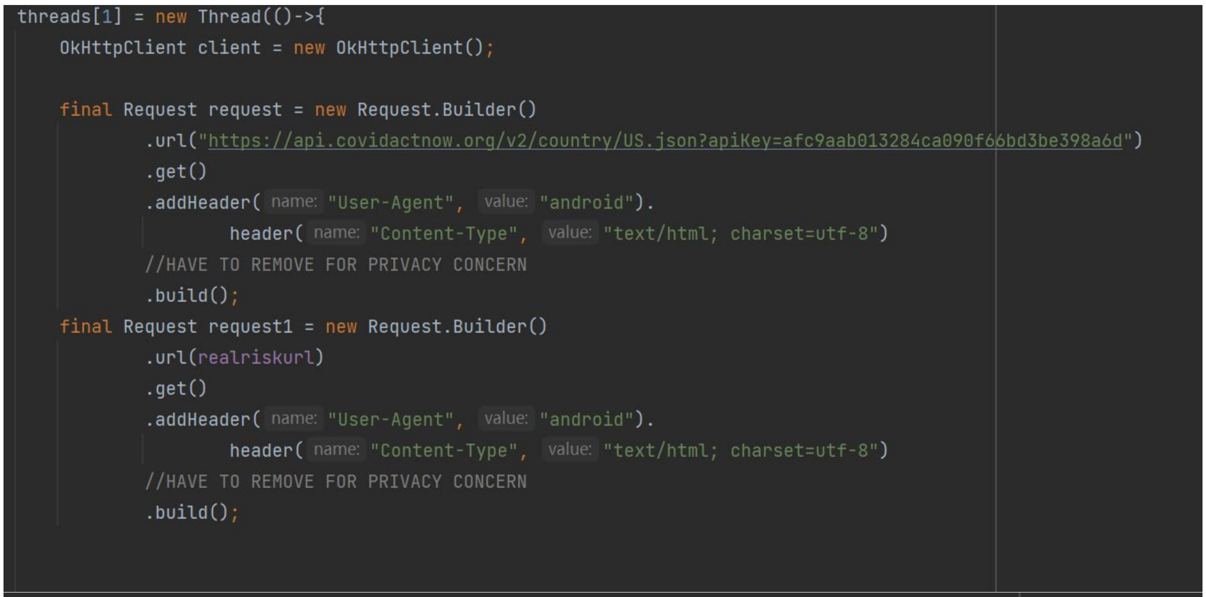
```java
if (ActivityCompat.checkSelfPermission( context: this, android.Manifest.permission.ACCESS_FINE_LOCATION)
        != PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission( context: this,
        android.Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
    ActivityCompat.requestPermissions( activity: this,
            new String[]{android.Manifest.permission.ACCESS_FINE_LOCATION,
                    Manifest.permission.ACCESS_COARSE_LOCATION},
            REQUEST_LOCATION_PERMISSION_CODE);
    new Handler().postDelayed(new Runnable() {
        @Override
        public void run() {
            Intent LaunchIntent = getPackageManager().getLaunchIntentForPackage(getApplication().getPackageName());
            LaunchIntent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            startActivity(LaunchIntent);
        }
    },  delayMillis: 3000);// 1秒钟后重启应用
} else {
    //System.out.println("11111111");
    fusedLocationClient = LocationServices.getFusedLocationProviderClient( activity: this);
    fusedLocationClient.getLastLocation()
            .addOnSuccessListener( activity: this, new OnSuccessListener<Location>() {

                @Override
                public void onSuccess(Location location) {

                    // Got last known location. In some rare situations this can be null.
                    if (location != null) {
                        // Logic to handle location object
                        System.out.println("1111111" + location);
                        String tmp =getAddress(location);
                        setState(tmp);
                    }
                }
            });
}
```



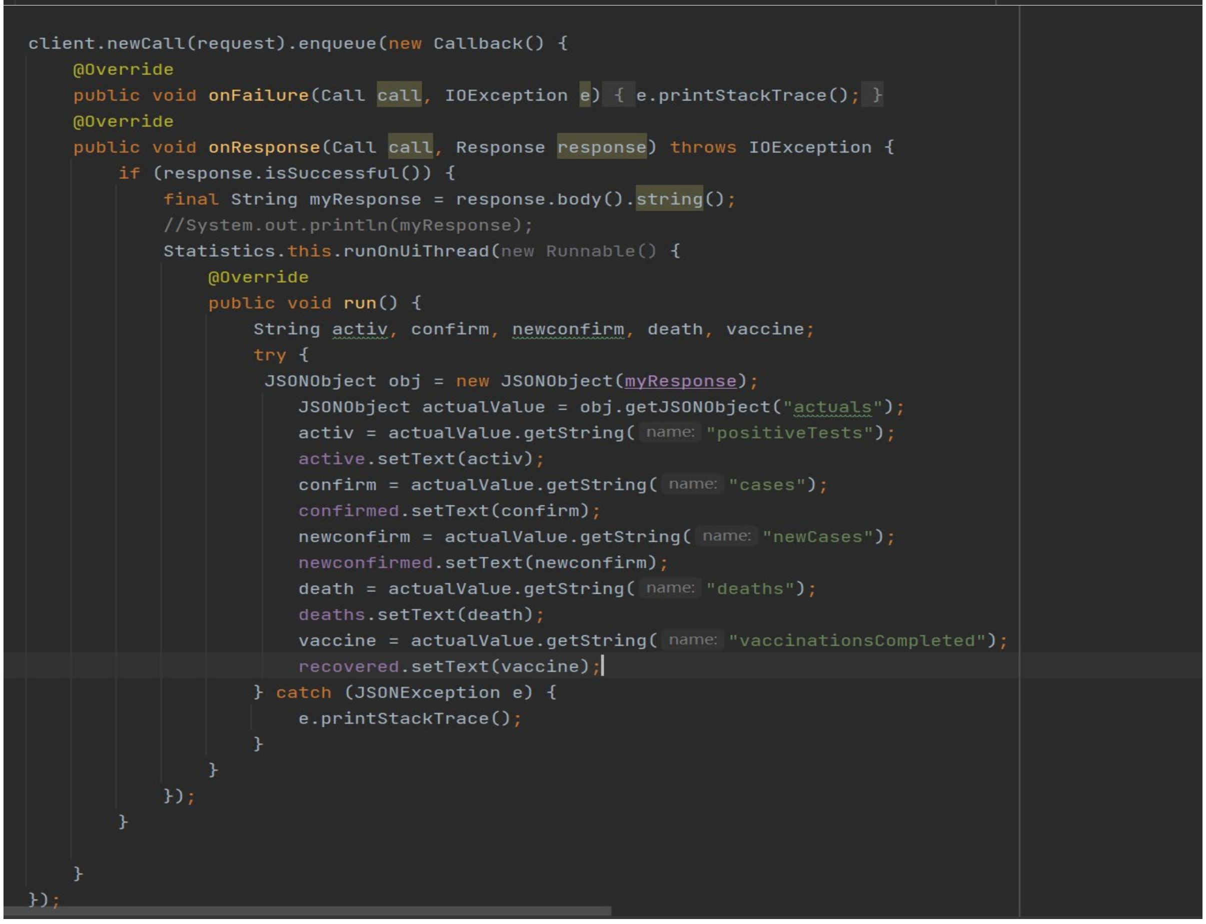
```java
private  String  getAddress(Location location) {
    List<Address> result = null;
    String realstate = null;
    try {
        if (location != null) {
            Geocoder gc = new Geocoder( context: this, Locale.getDefault());
            result = gc.getFromLocation(location.getLatitude(),
                    location.getLongitude(),  maxResults: 1);
            //            val geoApiCtx = GeoApiContext.Builder()
            //                    .apiKey(Util.readGoogleApiKey(location))
            //                    .build()
            //System.out.println(result.getClass());
            // Toast.makeText(this, "address info: "+result.toString(), Toast.LENGTH_LONG).show();
            Log.v( tag: "TAG",  msg: "address: "+result.toString());
            //System.out.println("11111"+ result.toString());
            Object[] address1 = result.toArray();
            //System.out.println(address1[0].toString());
            String adress = (result.get(0).getAddressLine( index: 0));
            String [] addressarray = adress.split( regex: ",");
            //System.out.println(addressarray[2]);
            String state = addressarray[2].toString();
            String [] statelist = state.split( regex: "\\s+");
            realstate = statelist[1];
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return realstate;
}
```

Statistic.java:

In this class, we focus on obtaining the current country's epidemic data and the current state's transmission coefficient. Using the Covid Act Now API. In this class, we use the OkHttp API to make network requests.



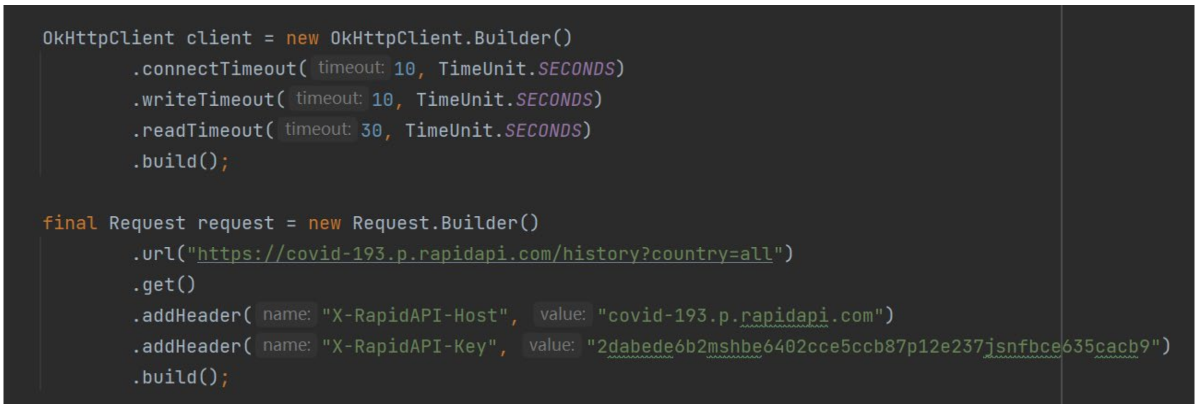
```java
threads[1] = new Thread(()->{
    OkHttpClient client = new OkHttpClient();

    final Request request = new Request.Builder()
            .url("https://api.covidactnow.org/v2/country/US.json?apiKey=afc9aab013284ca090f66bd3be398a6d")
            .get()
            .addHeader( name: "User-Agent",  value: "android").
                    header( name: "Content-Type",  value: "text/html; charset=utf-8")
            //HAVE TO REMOVE FOR PRIVACY CONCERN
            .build();
    final Request request1 = new Request.Builder()
            .url(realriskurl)
            .get()
            .addHeader( name: "User-Agent",  value: "android").
                    header( name: "Content-Type",  value: "text/html; charset=utf-8")
            //HAVE TO REMOVE FOR PRIVACY CONCERN
            .build();
```



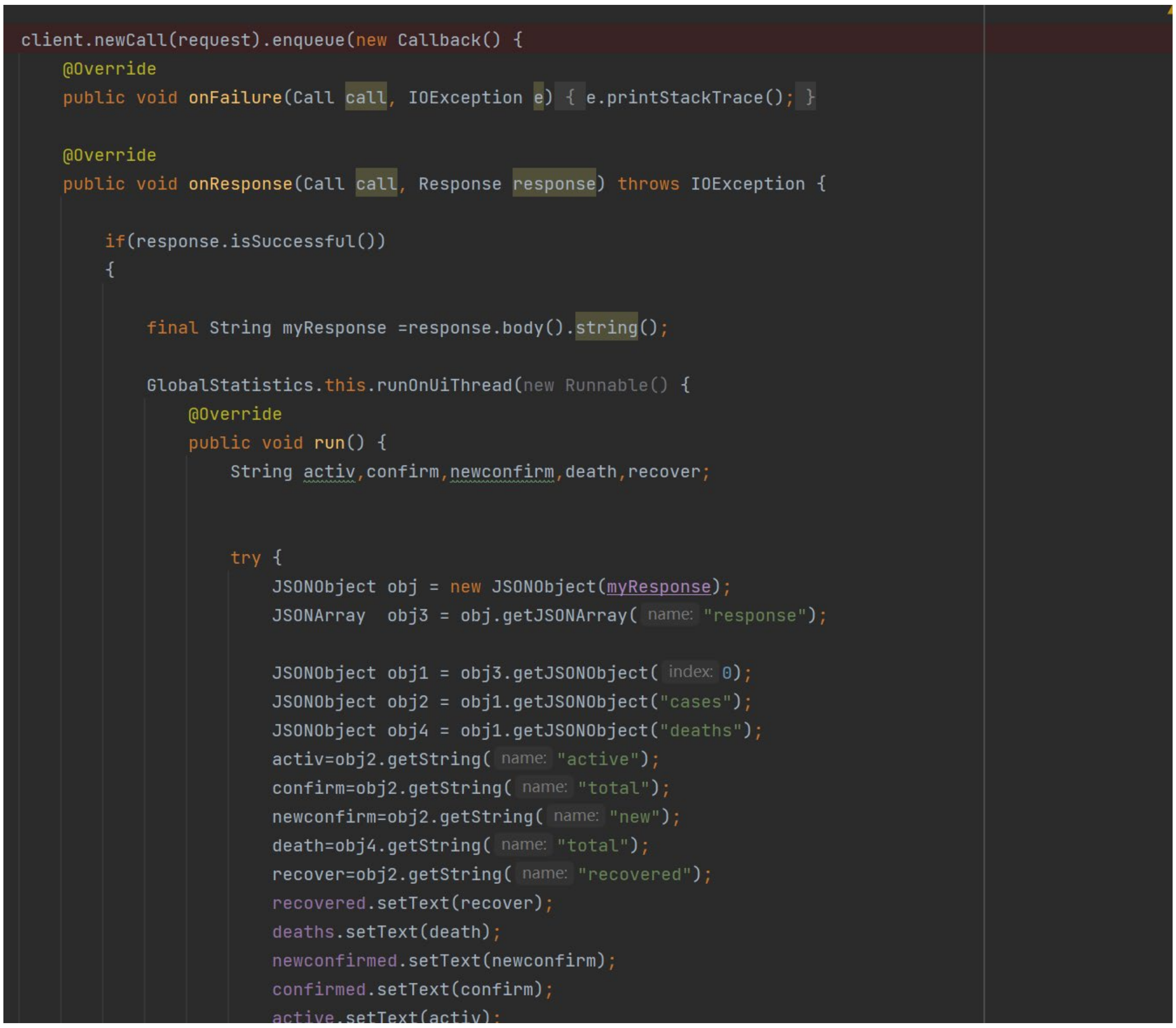
```java
client.newCall(request).enqueue(new Callback() {
    @Override
    public void onFailure(Call call, IOException e) { e.printStackTrace(); }
    @Override
    public void onResponse(Call call, Response response) throws IOException {
        if (response.isSuccessful()) {
            final String myResponse = response.body().string();
            //System.out.println(myResponse);
            Statistics.this.runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    String activ, confirm, newconfirm, death, vaccine;
                    try {
                     JSONObject obj = new JSONObject(myResponse);
                        JSONObject actualValue = obj.getJSONObject("actuals");
                        activ = actualValue.getString( name: "positiveTests");
                        active.setText(activ);
                        confirm = actualValue.getString( name: "cases");
                        confirmed.setText(confirm);
                        newconfirm = actualValue.getString( name: "newCases");
                        newconfirmed.setText(newconfirm);
                        death = actualValue.getString( name: "deaths");
                        deaths.setText(death);
                        vaccine = actualValue.getString( name: "vaccinationsCompleted");
                        recovered.setText(vaccine);
                    } catch (JSONException e) {
                        e.printStackTrace();
                    }
                }
            });
        }

    }
});
```

GlobalStatistics.java

Same as statistic.java, but this time we are requesting global covid data



```
OkHttpClient client = new OkHttpClient.Builder()
        .connectTimeout( timeout: 10, TimeUnit.SECONDS)
        .writeTimeout( timeout: 10, TimeUnit.SECONDS)
        .readTimeout( timeout: 30, TimeUnit.SECONDS)
        .build();

final Request request = new Request.Builder()
        .url("https://covid-193.p.rapidapi.com/history?country=all")
        .get()
        .addHeader( name: "X-RapidAPI-Host",  value: "covid-193.p.rapidapi.com")
        .addHeader( name: "X-RapidAPI-Key",  value: "2dabede6b2mshbe6402cce5ccb87p12e237jsnfbce635cacb9")
        .build();
```



```
client.newCall(request).enqueue(new Callback() {
    @Override
    public void onFailure(Call call, IOException e) { e.printStackTrace(); }

    @Override
    public void onResponse(Call call, Response response) throws IOException {

        if(response.isSuccessful())
        {

            final String myResponse =response.body().string();

            GlobalStatistics.this.runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    String activ,confirm,newconfirm,death,recover;


                    try {
                        JSONObject obj = new JSONObject(myResponse);
                        JSONArray  obj3 = obj.getJSONArray( name: "response");

                        JSONObject obj1 = obj3.getJSONObject( index: 0);
                        JSONObject obj2 = obj1.getJSONObject("cases");
                        JSONObject obj4 = obj1.getJSONObject("deaths");
                        activ=obj2.getString( name: "active");
                        confirm=obj2.getString( name: "total");
                        newconfirm=obj2.getString( name: "new");
                        death=obj4.getString( name: "total");
                        recover=obj2.getString( name: "recovered");
                        recovered.setText(recover);
                        deaths.setText(death);
                        newconfirmed.setText(newconfirm);
                        confirmed.setText(confirm);
                        active.setText(activ);
```

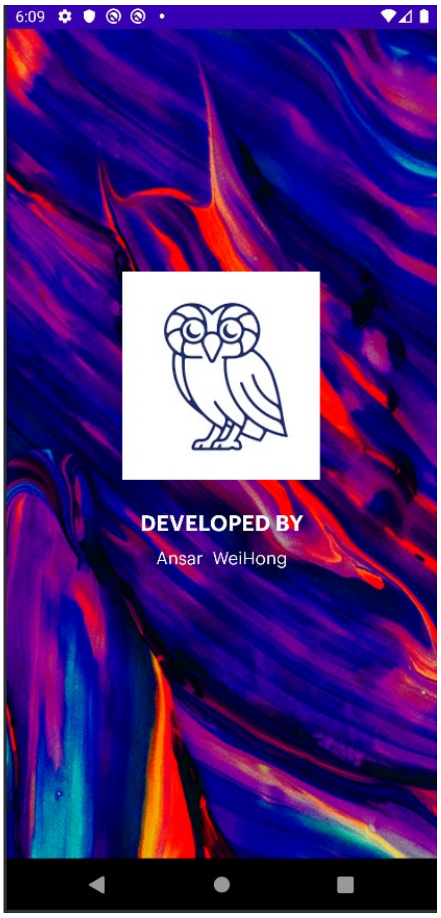
6 GitHub repositories

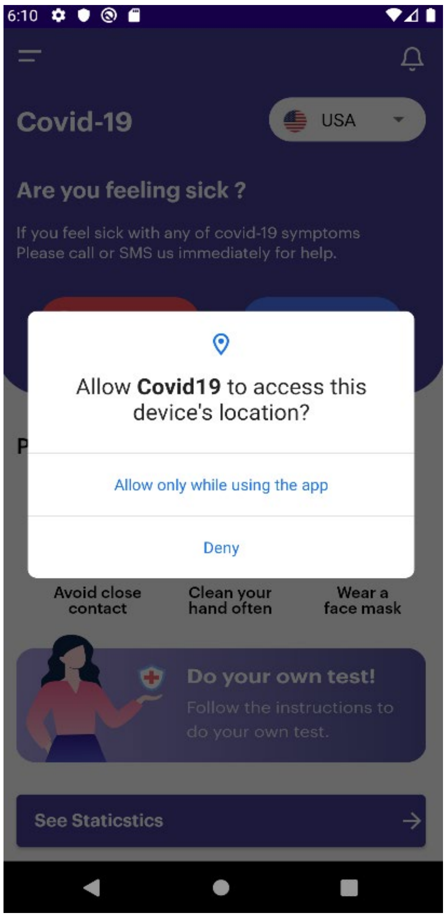
Application and Code Link :

https://github.com/ansarrice/covid19-comp590/tree/master

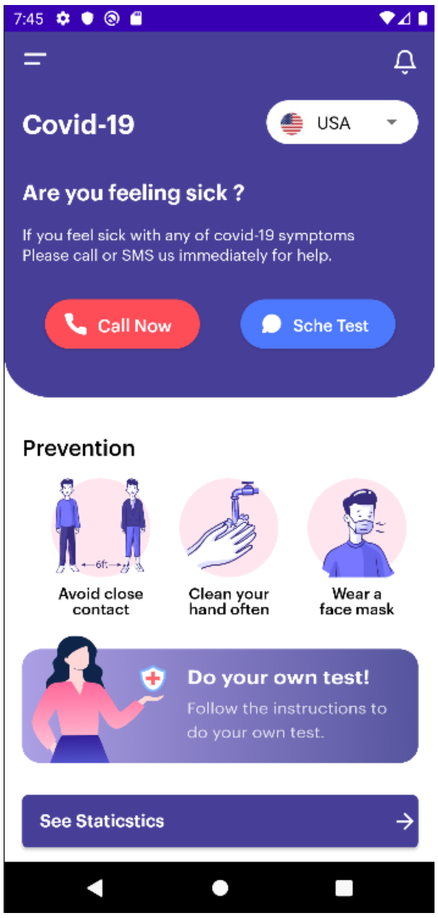
7 Graphics

Next, let's introduce the actual image of the APP running on the physical machine.





6:10
Covid-19
USA
Are you feeling sick ?
If you feel sick with any of covid-19 symptoms
Please call or SMS us immediately for help.
Allow **Covid19** to access this device's location?
Allow only while using the app
Deny
Avoid close contact
Clean your hand often
Wear a face mask
Do your own test!
Follow the instructions to do your own test.
See Staticstics

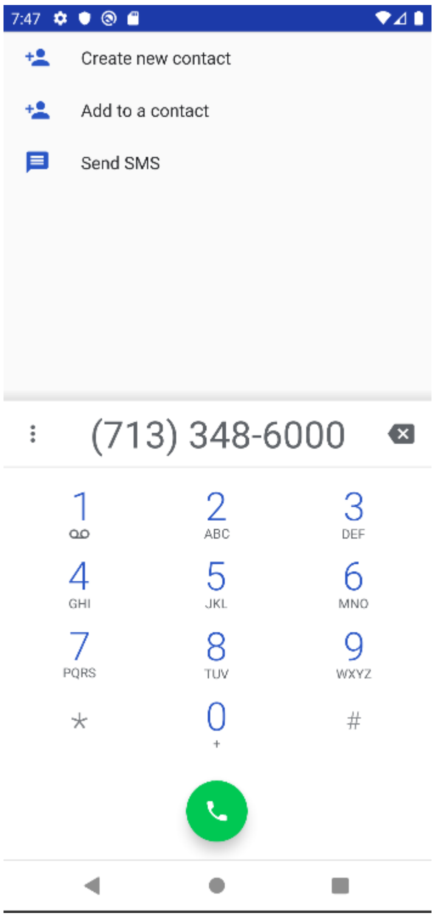


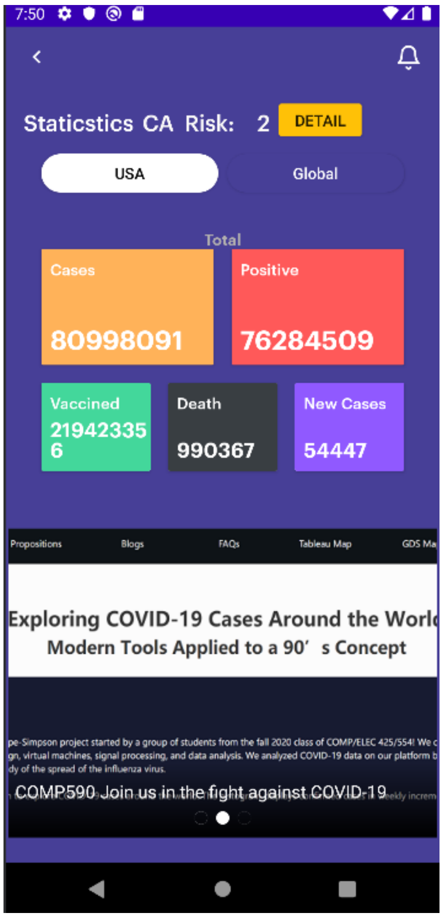
Are you feeling sick ?

If you feel sick with any of covid-19 symptoms
Please call or SMS us immediately for help.

Do your own test!

Follow the instructions to do your own test.





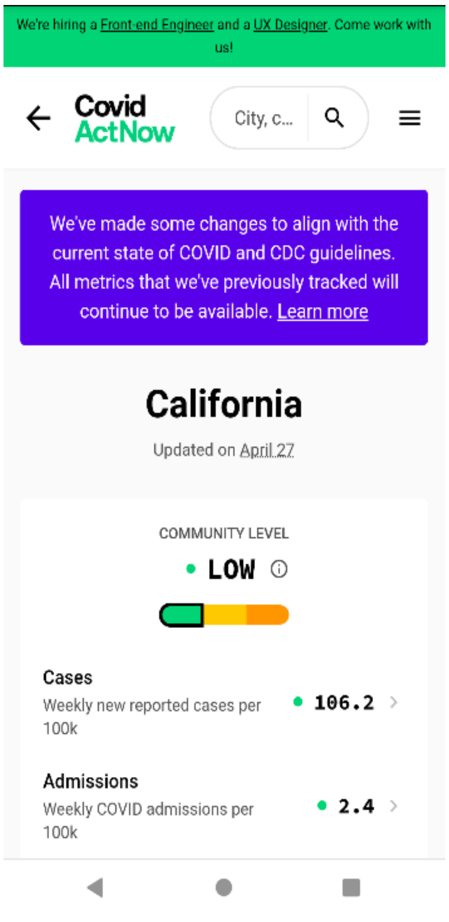
Cases	Positive
80998091	76284509

Vaccined	Death	New Cases
219423356	990367	54447

Exploring COVID-19 Cases Around the Worl
Modern Tools Applied to a 90's Concept

pe-Simpson project started by a group of students from the fall 2020 class of COMP/ELEC 425/554! We c
gn, virtual machines, signal processing, and data analysis. We analyzed COVID-19 data on our platform b
dy of the spread of the influenza virus.

COMP590 Join us in the fight against COVID-19



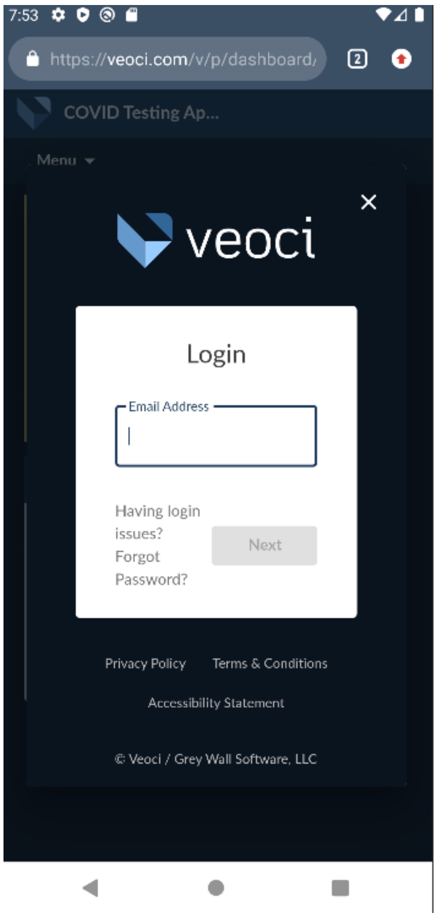
We're hiring a Front-end Engineer and a UX Designer. Come work with us!

We've made some changes to align with the current state of COVID and CDC guidelines. All metrics that we've previously tracked will continue to be available. Learn more

California

Updated on April 27

Cases Weekly new reported cases per 100k	● 106.2
Admissions Weekly COVID admissions per 100k	● 2.4


7:53

https://veoci.com/v/p/dashboard/

COVID Testing Ap...

Menu

veoci

Login

Email Address

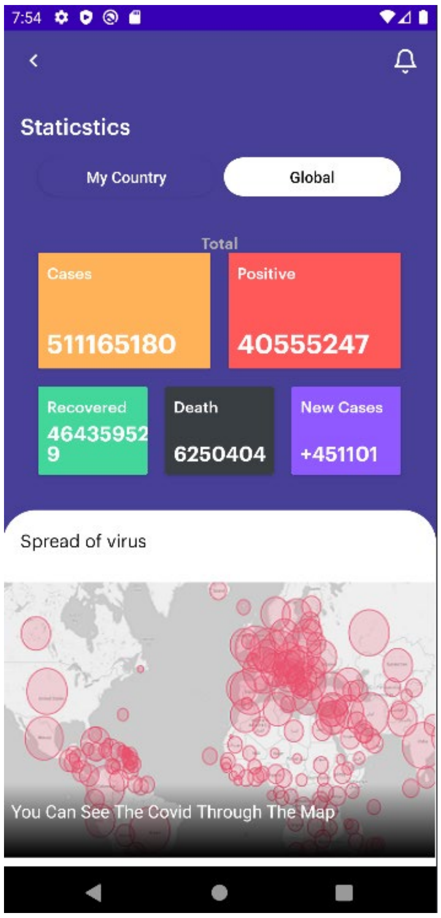
Having login issues? Forgot Password?

Next

Privacy Policy Terms & Conditions

Accessibility Statement

© Veoci / Grey Wall Software, LLC


7:54

Staticstics

My Country Global

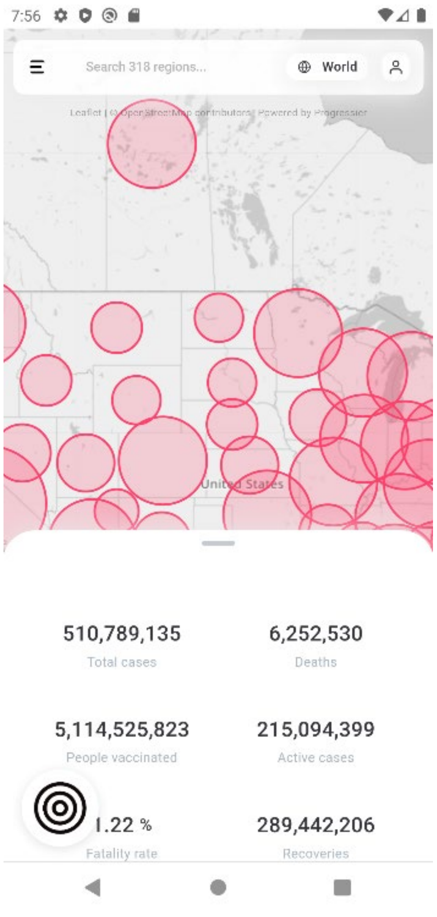
Total

Cases	Positive
511165180	40555247

Recovered	Death	New Cases
464359529	6250404	+451101

Spread of virus

You Can See The Covid Through The Map

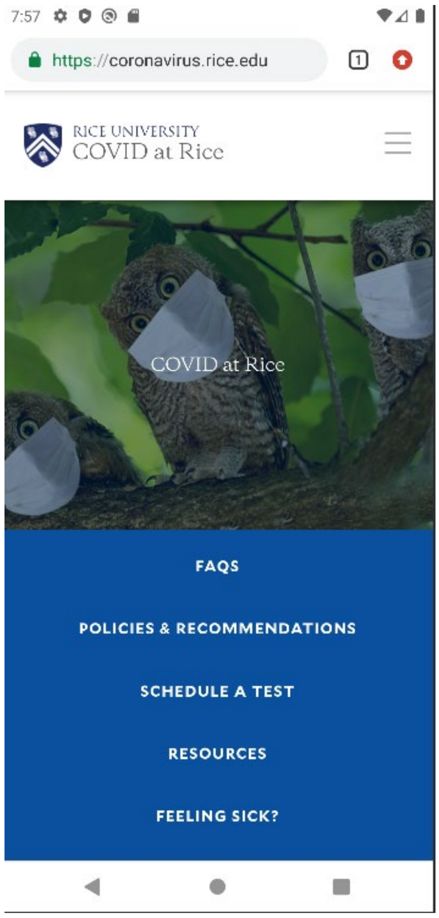

7:56

Search 318 regions... World

Leaflet | © OpenStreetMap contributors | Powered by Progressier

United States

510,789,135	6,252,530
Total cases	Deaths
5,114,525,823	215,094,399
People vaccinated	Active cases
1.22 %	289,442,206
Fatality rate	Recoveries


7:57

https://coronavirus.rice.edu

RICE UNIVERSITY
COVID at Rice

COVID at Rice

FAQS

POLICIES & RECOMMENDATIONS

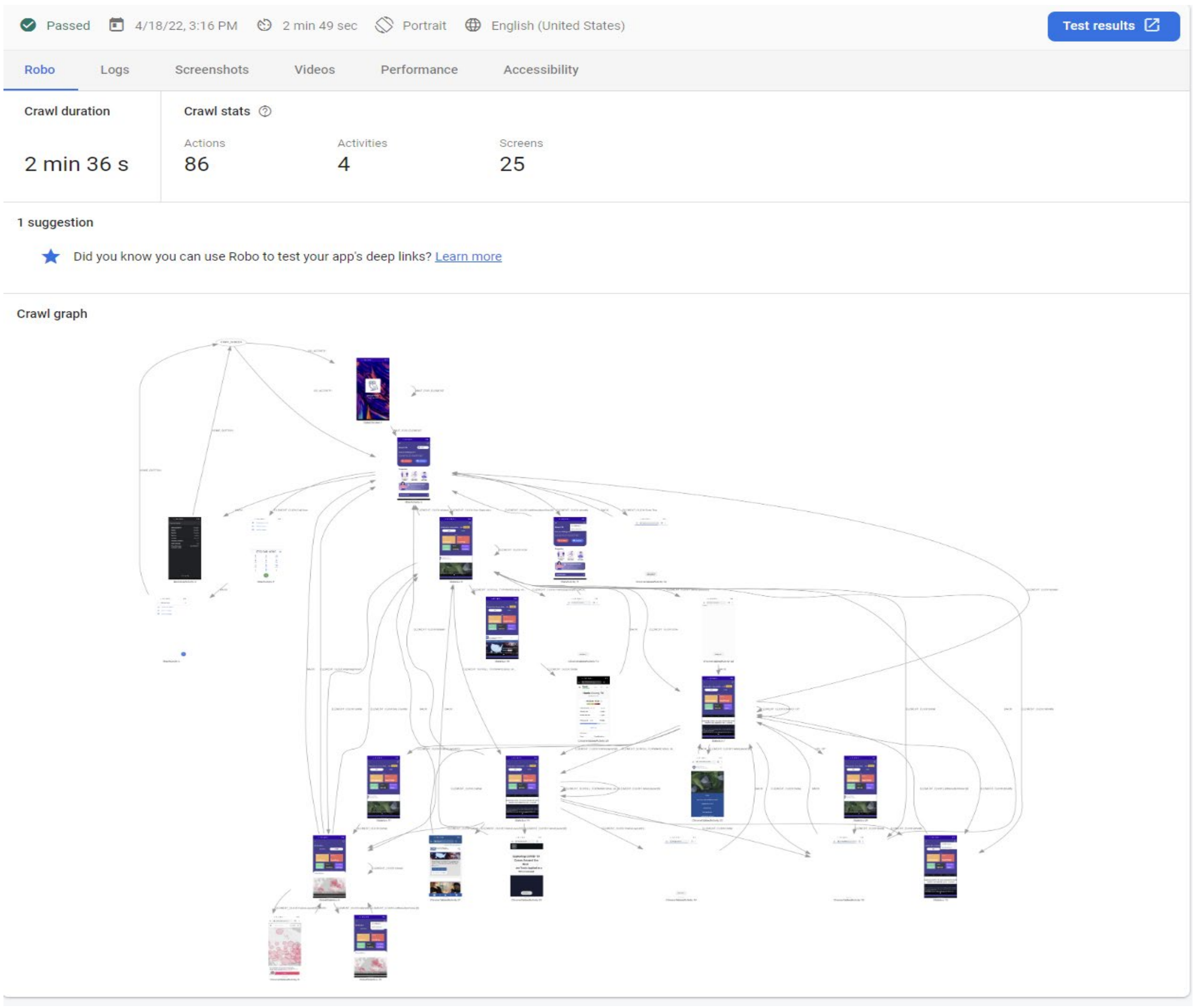
SCHEDULE A TEST

RESOURCES

FEELING SICK?

8 Conclusion(Application Test)

In this part, we conducted a third-party test reference. Testing the Covid19 App on different models and different Android versions concluded that the App has a lot of room for improvement in scalability.


Passed 4/18/22, 3:16 PM 2 min 49 sec Portrait English (United States) Test results

Robo Logs Screenshots Videos Performance Accessibility

Crawl duration

2 min 36 s

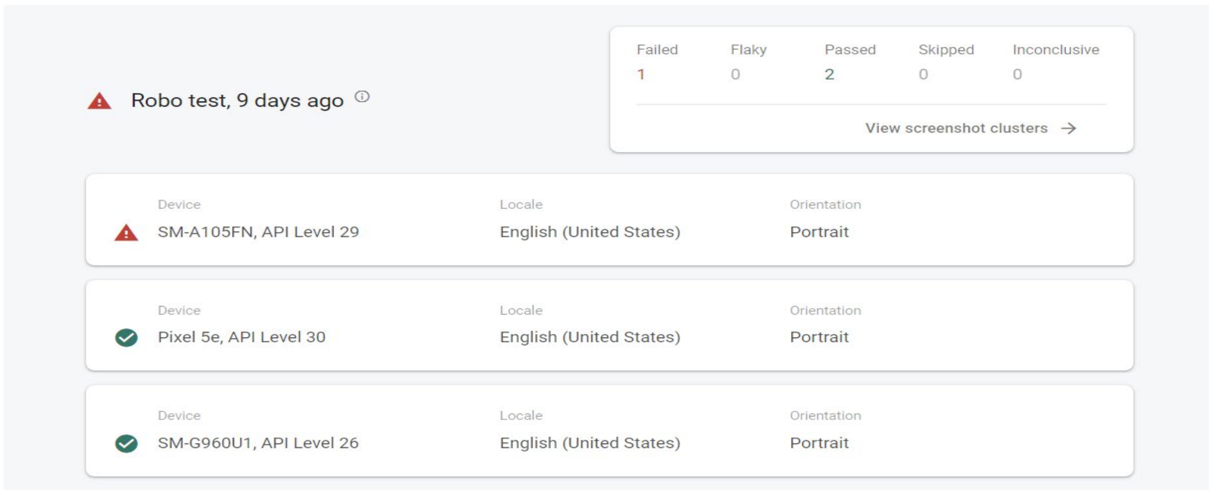
Crawl stats

Actions	Activities	Screens
86	4	25

1 suggestion

Did you know you can use Robo to test your app's deep links? Learn more

Crawl graph


Robo test, 9 days ago

Failed	Flaky	Passed	Skipped	Inconclusive
1	0	2	0	0

View screenshot clusters →

Device	Locale	Orientation
SM-A105FN, API Level 29	English (United States)	Portrait
Pixel 5e, API Level 30	English (United States)	Portrait
SM-G960U1, API Level 26	English (United States)	Portrait

Although the application has met the user's requirements for obtaining the most basic epidemic data, as people understand the virus, there are more valuable data for people to refer to, so the program should continue to be updated.

With the development of the mobile Internet, people pay more attention to their dependence on mobile phones, so our original intention is to provide more information about the epidemic to users through the Phone. And through practice and user feedback, we also realize that in addition to providing effective information, we also need to provide users with more practical functions, such as online appointments for nucleic acid testing, such as emergency calls, etc.

9 References

Google Android developer guides.

Link: https://developer.android.com/guide

10 Reproducibility

Our team developed the Covid19 Application during the learning of Android development this semester. Covid19 App is not perfect, and there may still be some bugs, so we will continue to optimize and improve it.

The App still has many, many good expansion directions in terms of functions. For example, we can add a user login function; and add a campus forum function, so that students in the school can exchange information about Covid19 on the App. And we can also add an online medical platform so that doctors on campus can provide students with some health knowledge online or provide some online medical help.

And as our understanding of the virus deepens and the epidemic data becomes more comprehensive, we have more sources of information, such as the Covid Act Now API, which provides us with many effective APIs to obtain more effective information, all of which can be The expansion of the APP, such as the analysis between the vaccination rate of the vaccine and the speed of the epidemic, and the analysis of the community transmission coefficient can become the main direction of our future research.