

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Mobile application development is the process to making software for smart phones and digital assistants, most commonly for Android and iOS. The software can be preinstalled on the device, downloaded from a mobile app store or accessed through a mobile web browser. The programming and markup languages used for this kind of software development include Java, Swift, C# and HTML5.

Mobile app development is rapidly growing. From retail, telecommunications and e-commerce to insurance, healthcare and government, organizations across industries must meet user expectations for real-time, convenient ways to conduct transactions and access information. Today, mobile devices and the mobile applications that unlock their value are the most popular way for people and businesses to connect to the internet. To stay relevant, responsive and successful, organizations need to develop the mobile applications that their customers, partners and employees demand. Yet mobile application development might seem daunting. Once you've selected the OS platform or platforms, you need to overcome the limitations of mobile devices and usher your app all the way past the potential hurdles of distribution. Fortunately, by following a few basic guidelines and best practices, you can streamline your application development journey. We can start explaining mobile development, which is not about building phone apps, though it is a huge part of it.

Actually, It's doing any reasonably development for any kind of mobile devices such as developing apps for phones, tablets, smart watches, and every form of wearable devices that run any kind of mobile operating system.

Mobile development presents a reasonably distinctive chance for a one-person development team to build an actual, usable, significant app end-to-end during a comparatively short period. However, mobile apps development represents more than just a chance for the solo-developer to create their own project as it is arguably the longer term of development, as mobile devices are getting larger and bigger parts of our lives.

1.2 PROBLEM STATEMENT

The rapid spread of the novel coronavirus and its resulting condition, COVID-19, has caught much of the world off-guard. The tragedy has yet to fully play out, but it is already clear that the crisis is thoroughly global in nature and that science is on the front lines in the fight against the virus. This includes medical professionals attempting to heal the sick at risk to their own health, public health officials tracking the virus and vigilantly urging such measures as social distancing to mitigate its spread, and researchers now engaged in the development of diagnostics, treatments and vaccines.

1.3 MOBILE APPLICATION DEVELOPMENT NEED & IMPORTANCE

App development is essential today for an online business. If you want to boost your sales using technology, app development is truly recommended for everyone. Here listed 4 reasons that you need to know how much it notable is.

- **Accessibility from Variety of Platforms**

You may gain access to nearly every type of online platforms by developing your apps. Develop apps will help reach into marketplaces as far reaching areas via Google Play, Blackberry, as well as Apple App Store, Symbian, and other internet marketplaces and through social media web sites just like Facebook or Myspace, Twitter, among others. Besides having the ability to mail data to clients, app growth possesses additional exclusive capabilities, which include coupon codes, evaluation of functions, and also force announcements. Your visitors within the quickest feasible valuable time, and also obtain an immediate answer, which assists you examine the advertising tool.

- **Targeted Audience**

It's simple to get obsessed with app ideas. I hear fantastic ideas daily, but generally they're strategies that originate around a work or function, instead of a particular target audience. The audience is very close to an afterthought, merely crucial while making the advertising program. The a lot more applications I style and also release, the a lot more I'm certain it's easier to begin with a particular audience and also produce suggestions depending on the requirements of this target audience.

- **Efficiency and Effectiveness**

Effectiveness and efficiency: Business management of internet based use is conducted with effectiveness and also performance. Can decrease their functional expenses whilst at the same time improving the effectiveness of their procedures. Can decrease their producing expenditures, which make your company environment friendly. Threatened functions offer customers the ability of handling their companies and never have to get the cost of employing an additional worker for carrying this out train.

Internet based functions will not need customers to install these types of on their hard disk drives that lead to the decrease of memory space. Moreover, any specific up-to-date variations can be found immediately for the customers. The dependable and also efficiently created internet based functions are created to make sure that they are suitable for all of the different internet browsers, working devices, and also equipment.

- **To engage with customers everywhere**

You can involve with your all types of customers everywhere by developing your apps system. Modern and updated apps perform multiple task in business as well as other site too. You should develop your apps system that is really user friendly and easy to access so that customer can contact anytime from anywhere they are.

CHAPTER 2

SYSTEM REQUIREMENT SPECIFICATION

2.1 SOFTWARE REQUIREMENTS

Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application.

The following are the software requirements for the application:

- Operating System: Windows 10
- Development Environment: Android Studio 4.2
- API: Java Development Kit (JDK) 7
- Core Language: Java, XML for Front-end.

2.2 HARDWARE REQUIREMENTS

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware.

- CPU: Intel processor with support for Intel VT-x (Intel 64)
- Cores: Dual-Core (Quad-Core recommended)
- RAM: minimum 4GB (>4GB recommended)
- Secondary Storage: 250GB hard disk space plus at least 1GB for Android SDK,
- Emulator System images, and caches.
- Screen resolution: 1366 x 800.

2.3 SOFTWARE TOOLS USED

- **Android Studio**

In recent times, Android became the world's most popular operating system for various reasons. As an Android programmer, I want to share what the Android Studio is? Android

Studio is an IDE for Google Android Development launched on 16th May 2013, during Google's I/O 2013 event. Android Studio contains all the Android tools to design, test, debug, and profile your application. The Android Studio uses Gradle to manage your project, a Build Automation Tool.

For developing your first app, you need to download Android Studio for your preferred platform (Windows®, Mac OS X, or Linux) from the Android developers site. Android Studio can develop and test your application on either a real device or an emulator.

Android Studio can be installed on Windows operating systems, OSX and Linux and is recommended by Google itself that the hardware must have at least 4 GB of memory and 1GB of free hard disk space, but we recommend that you have more memory because it was noted that Android Studio is still a little slow. You must have Java installed on the machine via the JDK (Java Development Kit), not the JRE, as it is usually installed, once to develop on Android is necessary for all Java development classes to be present on the machine.

Android Studio has many exciting features that can help you to develop your Android application like:

- Powerful code editor with smart editing and code re-factoring.
- Emulator to show your code output in various resolutions, including Nexus 4, Nexus 7, Nexus 10, and many other android phones.
- Gradle based build support.
- Maven Support.
- Template-based wizards.
- Dracula Theme Environment to enjoy your coding experience.
- You can experience all the awesome features by using Android Studio in-hand.

The user interface

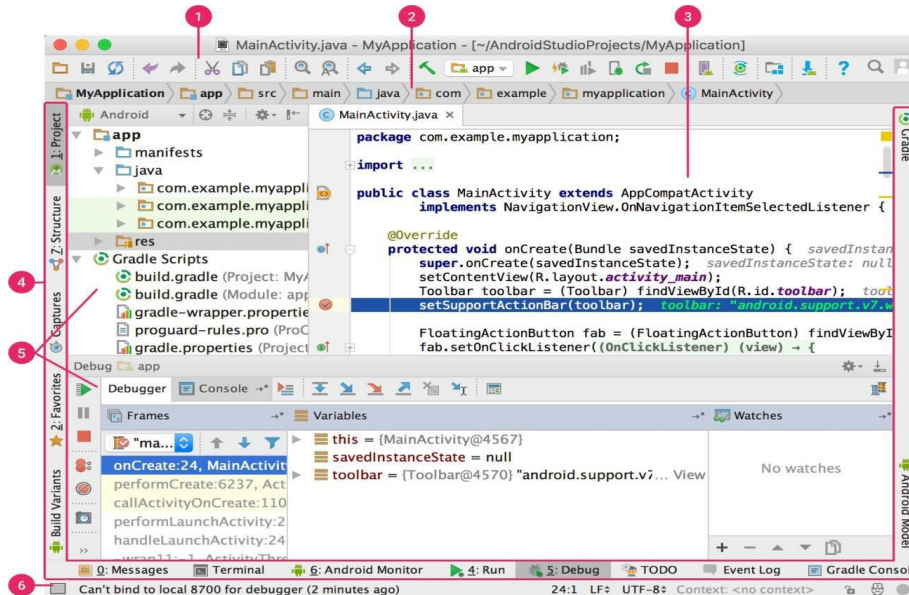


Figure 1: The Android Studio main window.

- **The toolbar** lets you carry out a wide range of actions, including running your app and launching Android tools.
- **The navigation bar** helps you navigate through your project and open files for editing. It provides a more compact view of the structure visible in the Project window.
- **The editor window** is where you create and modify code. Depending on the current file type, the editor can change.
- **The tool window bar** runs around the outside of the IDE window and contains the buttons that allow you to expand or collapse individual tool windows.
- **The tool windows** give you access to specific tasks like project management, search, version control, and more.
- **The status bar** displays the status of your project and the IDE itself, as well as any warnings or messages

CHAPTER 3

SYSTEM ANALYSIS

3.1 API

COVID19-India API

- Aggregated sheets provide aggregated data at the district/state levels in csv format.
- V4 json endpoints. These are the json apis that are used by the website to show all the statistics on the site. These can be used by developers and analysts who have knowledge of json parsing (recommended approach). All our v4 endpoints are actively developed and in use since this serves the frontend view Documentation for the same.
- Latest data from the google sheet (10-20 minutes delayed) is available through the latest end-point. These are present under the raw files section below. (Not recommended since the number of files is huge and there is no additional information present in these as compared to the above mentioned endpoints.)

API	Link to CSV	Description
case_time_series	https://api.covid19india.org/csv/latest/case_time_series.csv	India level timeseries for Confirmed, Recovered and Deceased cases
states	https://api.covid19india.org/csv/latest/states.csv	Statewise timeseries of Confirmed, Recovered and Deceased numbers.
districts	https://api.covid19india.org/csv/latest/districts.csv	Districtwise timeseries of Confirmed, Recovered and Deceased numbers.
state_wise_daily	https://api.covid19india.org/csv/latest/state_wise_daily.csv	Statewise per day delta of Confirmed, Recovered and Deceased numbers.
state_wise	https://api.covid19india.org/csv/latest/state_wise.csv	Statewise cumulative numbers till date.

COVID 19 TRACKER

district_wise	https://api.covid19india.org/csv/latest/district_wise.csv	Districtwise Cumulative numbers till date.
statewise_tested_numbers_data	https://api.covid19india.org/csv/latest/statewise_tested_numbers_data.csv	Number of tests conducted in each state, ventilators ,hospital bed occupy reported in state bulletins
tested_numbers_icmr_data	https://api.covid19india.org/csv/latest/tested_numbers_icmr_data.csv	Number of tests reported by ICMR
icmr_labs_state_wise	https://api.covid19india.org/csv/latest/icmr_labs_statewise.csv	Number of Labs in each state as per ICMR
sources_list	https://api.covid19india.org/csv/latest/sources_list.csv	List of sources that we are using.
rtPCR_samples_collected	http://api.covid19india.org/csv/latest/icmr_rtPCR_tests_daily.csv	Number of RTPCR samples collected statewise in ICMR Application

3.2. Data Flow Diagram

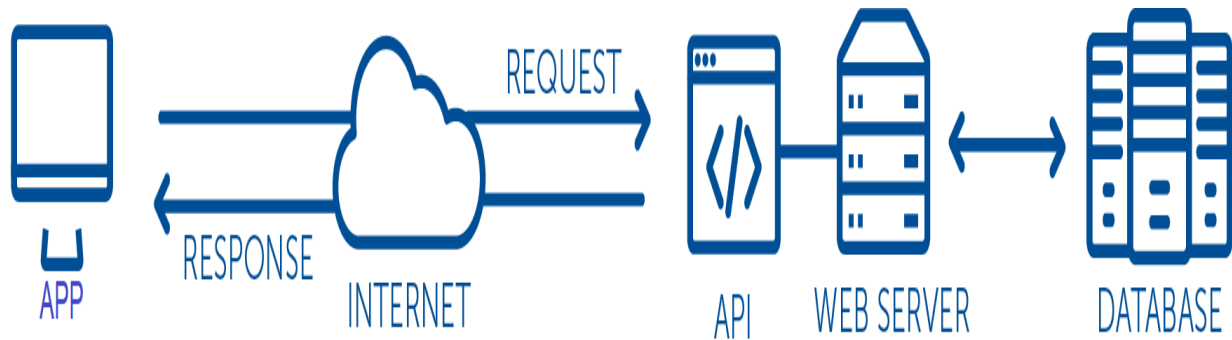


Figure 2: Data Flow Diagram

An API is a set of routines, protocols, and tools developers use to build software applications and to specify how software components should interact. In simple terms, an API is a messenger that sends a request to a provider you want data from and then delivers the response back to you. Think of an API as an intermediary that allows two programs to talk to each other.

In web development, an API allows applications to “piggyback” on other services.

To make communication and data sharing possible, an API exposes limited parts of a program’s interface so a specific set of features can be accessed and used by other programs. The applications are able to share data and take actions on each other without requiring Developers to share their entire code.

- **FIREBASE**

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client. When you build cross-platform apps with our iOS, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data.

CHAPTER 4

SYSTEM DESIGN

4.1 XML

XML (Extensible Markup Language) is a markup language similar to HTML, but without predefined tags to use. Instead, you define your own tags designed specifically for your needs. This is a powerful way to store data in a format that can be stored, searched, and shared. Most importantly, since the fundamental format of XML is standardized, if you share or transmit XML across systems or platforms, either locally or over the internet, the recipient can still parse the data due to the standardized XML syntax. There are many languages based on XML, including XHTML, MathML, SVG, XUL, XBL, RSS, and RDF. You can also define your own.

4.2. DESCRIPTION

- **ScrollView**

A view group that allows the view hierarchy placed within it to be scrolled. Scroll view may have only one direct child placed within it. To add multiple views within the scroll view, make the direct child you add a view group, for example LinearLayout, and place additional views within that LinearLayout. Scroll view supports vertical scrolling only. For horizontal scrolling, use HorizontalScrollView instead. Never add a RecyclerView or ListView to a scroll view. Doing so results in poor user interface performance and a poor user experience

- **CardView**

CardView uses elevation property on Lollipop for shadows and falls back to a custom emulated shadow implementation on older platforms. Due to expensive nature of rounded corner clipping, on platforms before Lollipop, CardView does not clip its children that intersect with rounded corners. Instead, it adds padding to avoid such intersection

- **CircleImageView**

As this is just a custom ImageView and not a custom Drawable or a combination of both, it can be used with all kinds of drawables, i.e. a PicassoDrawable from Picasso or other non-standard drawables (needs some testing though).

- **TextView**

A user interface element that displays text to the user. To provide user-editable text, see EditText.

- **Imageview**

Displays image resources, for example Bitmap or Drawable resources. ImageView is also commonly used to apply tints to an image and handle image scaling

- **Edittext**

A user interface element for entering and modifying text. When you define an edit text widget, you must specify the R.styleable.TextView_inputType attribute.

- **RecyclerView**

A flexible view for providing a limited window into a large data set.

- **NestedScrollView**

NestedScrollView is just like ScrollView, but it supports acting as both a nested scrolling parent and child on both new and old versions of Android. Nested scrolling is enabled by default.

- **EazeGraph**

EazeGraph is an Android library for creating beautiful and fancy charts. Its main goal was to create a lightweight library which is easy to use and highly customizable with an "up-to-date"-look. Currently 4 different chart types are available

- **View**

An XML view is one of the predefined view types that are available in OpenUI5. The XML view type is defined in an XML file. xml or as an XML string. ... The file name and the

COVID 19 TRACKER

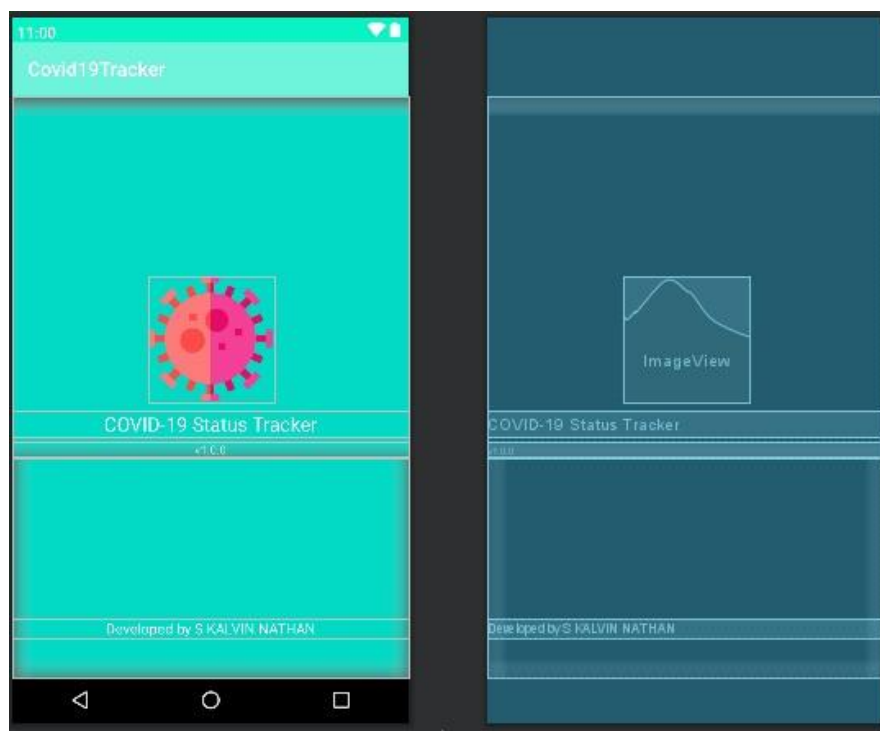
folder structure together specify the name of the view that equals the OpenUI5 module name within the require/declare concept

- **Others**

Item, Resources, String, Color, Version, Uses-permission, Application, Intent-filter, Category, Action, Relativelayout, Linearlayout, Swiperefreshlayout, Gridlayout, Org.eazegraph.lib.charts.piechart, Gradient, Aapt:attr, Vector, Gradient, Path, Shape.

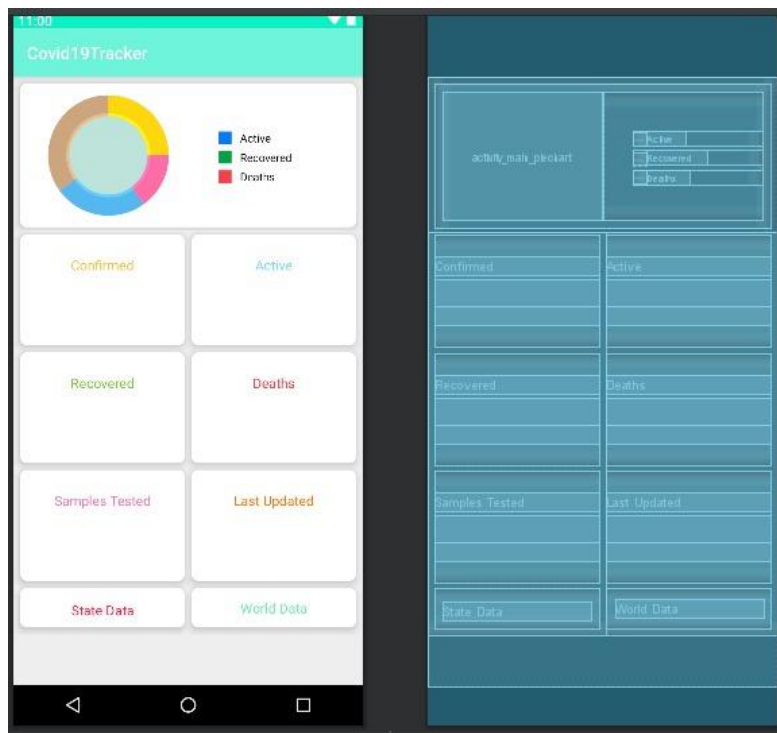
4.3 XML DESIGN

activity_splash

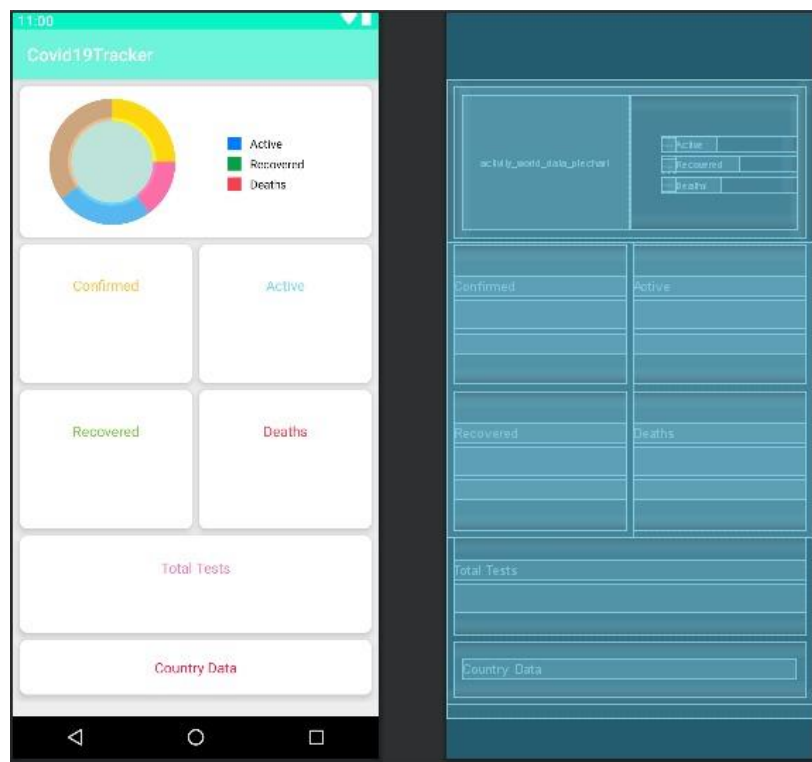


COVID 19 TRACKER

activity_main

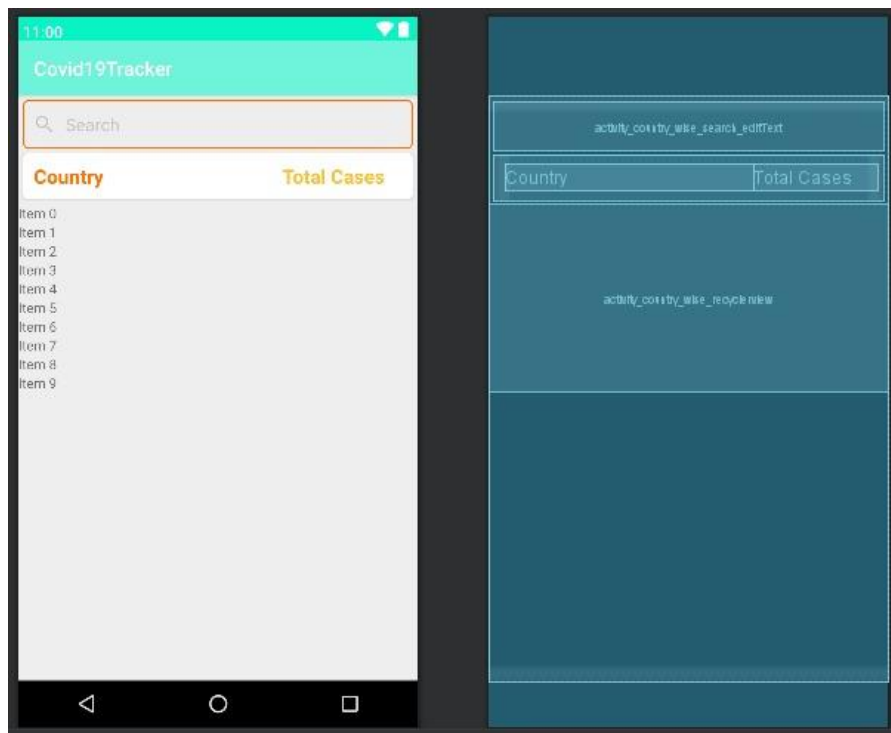


activity_world_data

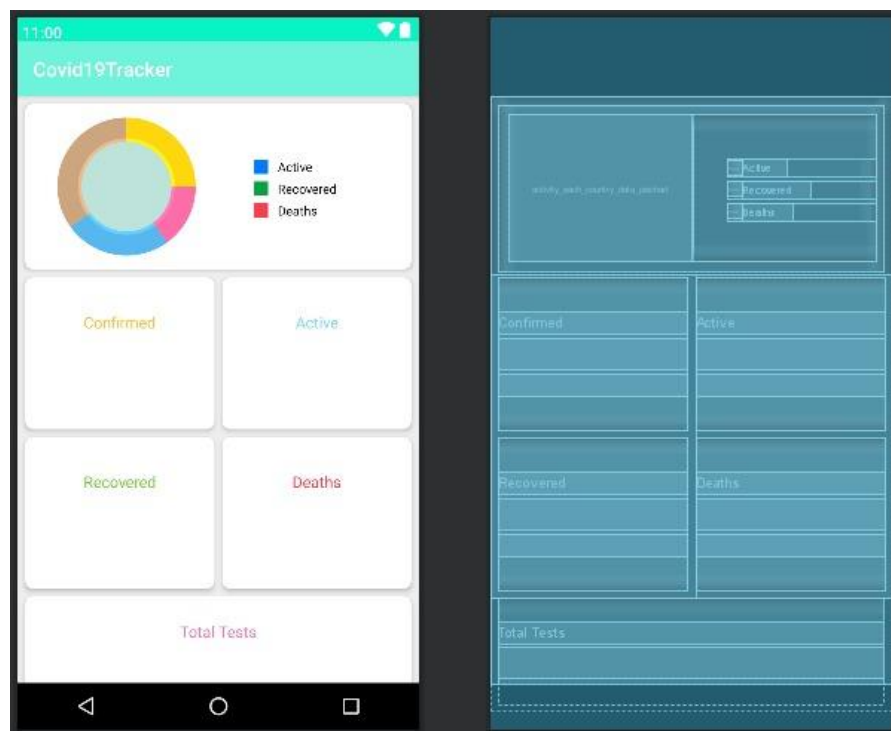


COVID 19 TRACKER

activity_country_wise_data

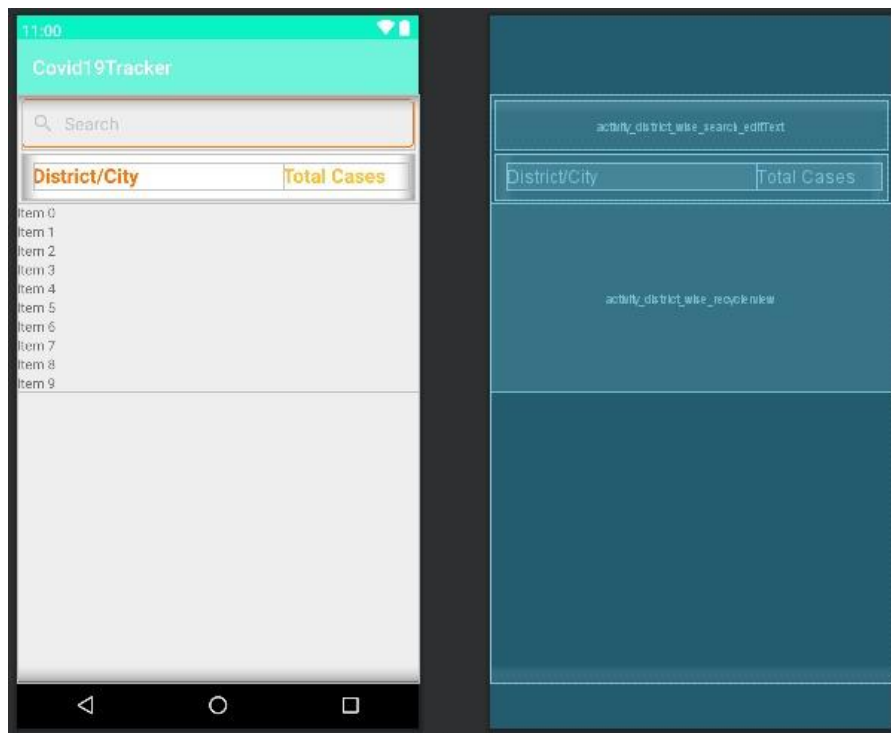


activity_each_country_data]

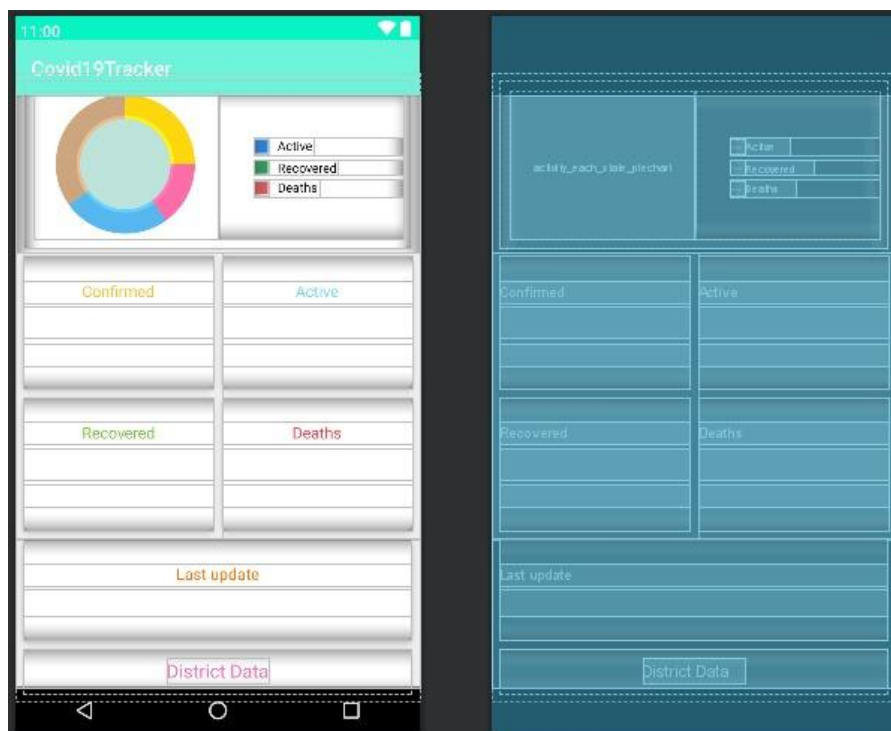


COVID 19 TRACKER

activity_district_wise_data

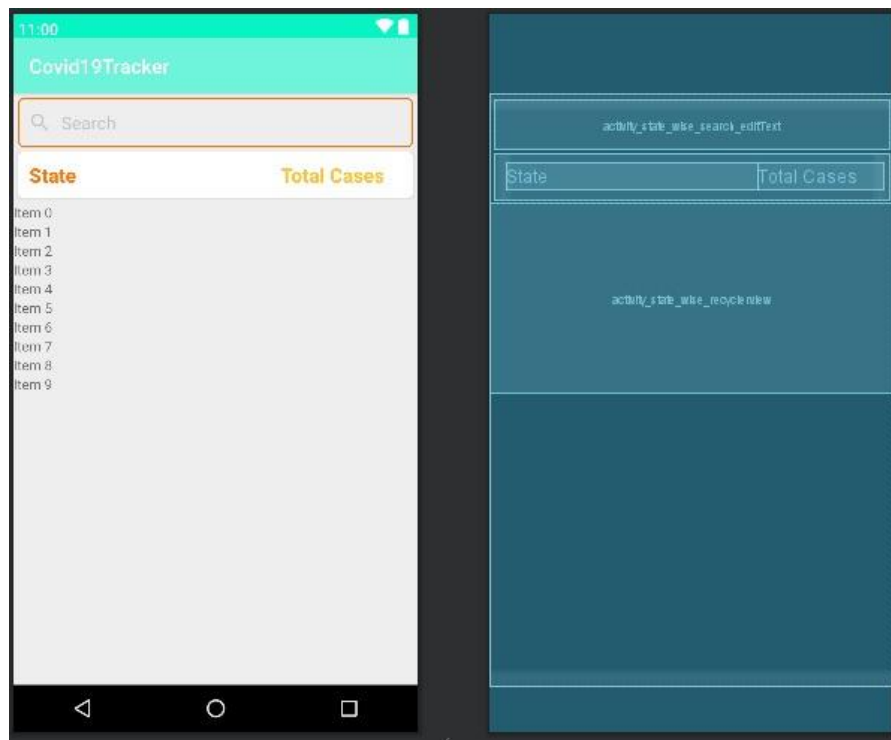


activity_each_state_data

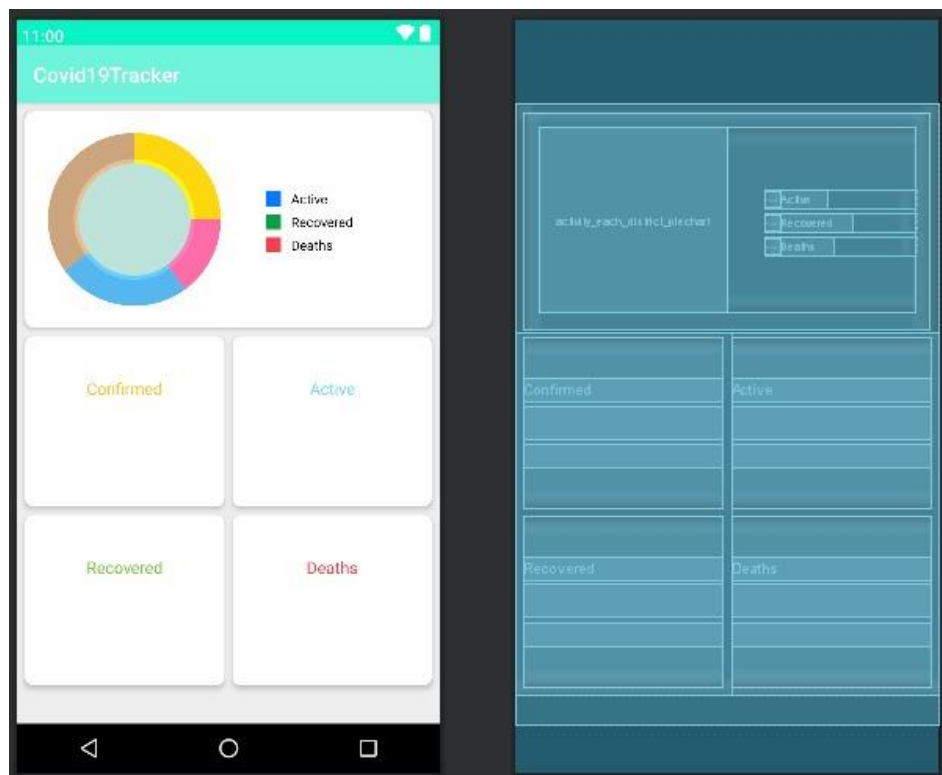


COVID 19 TRACKER

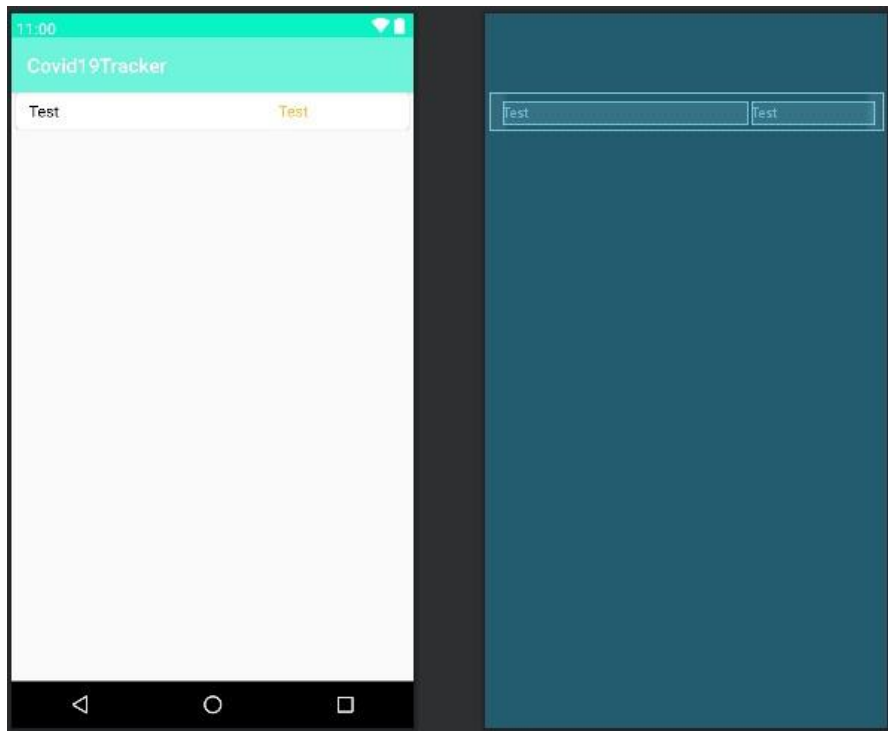
activity_state_wise_data



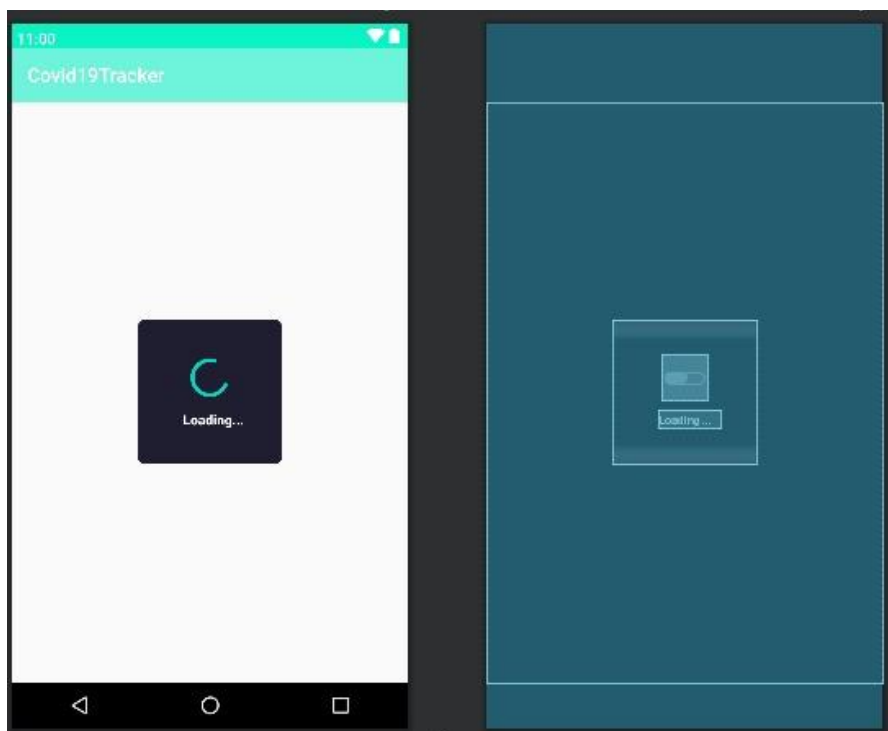
activity_each_district_data



layout_state_wise

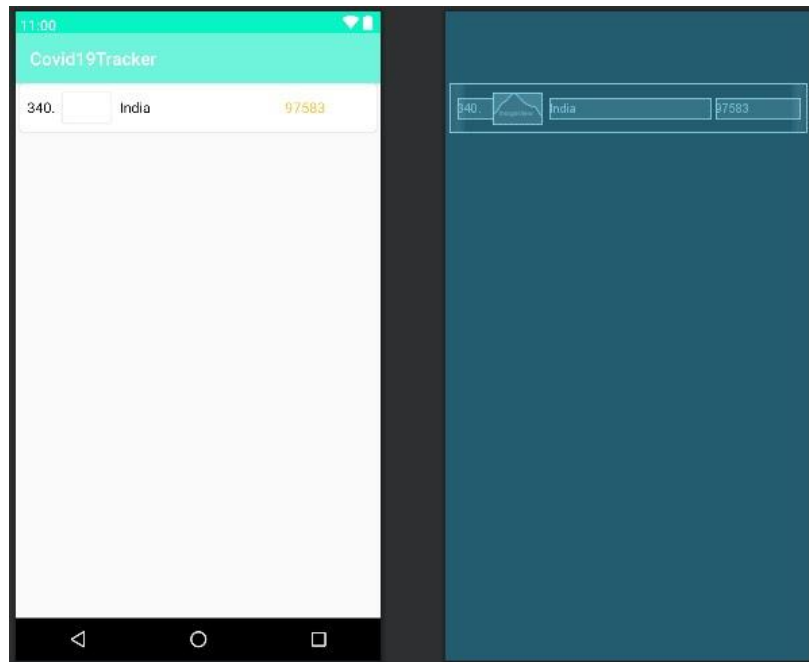


progress_dialog

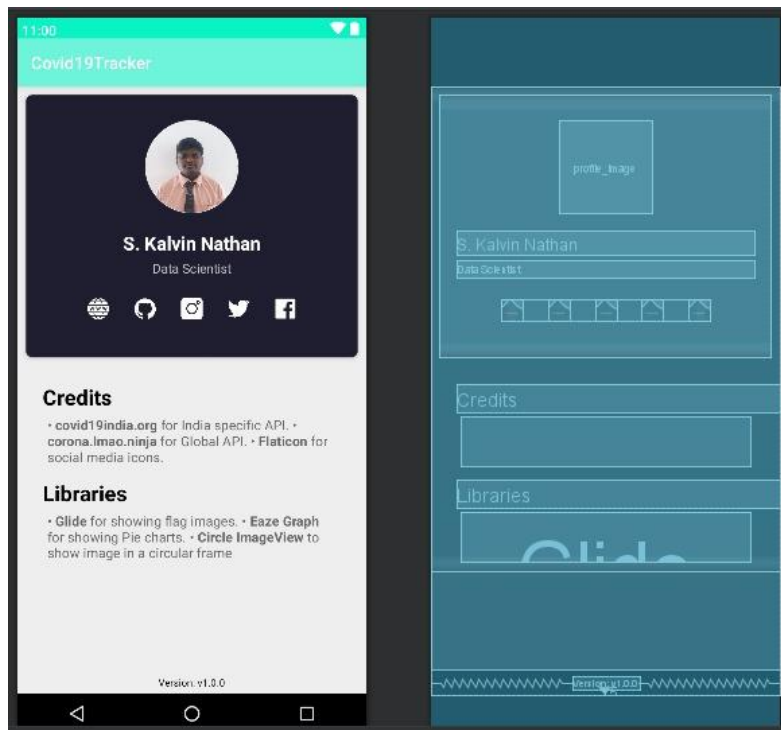


COVID 19 TRACKER

layout_country_wise



activity_about



CHAPTER 5

IMPLEMENTATION

5.1. JAVA

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let application developers *write once, run anywhere* (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages. As of 2019, Java was one of the most popular programming languages in use according to GitHub, particularly for client-server web applications, with a reported 9 million developers.

5.2. DESCRIPTION

- **Void onCreate(Bundle savedInstanceState)**

When an Activity first call or launched then onCreate(Bundle savedInstanceState) method is responsible to create the activity. When ever orientation(i.e. from horizontal to vertical or vertical to horizontal) of activity gets changed or when an Activity gets forcefully terminated by any Operating System then savedInstanceState i.e. object of Bundle Class will save the state of an Activity. After Orientation changed then onCreate(Bundle savedInstanceState) will call and recreate the activity and load all data from savedInstanceState.

- **Void Window.open()**

he `Window` interface's `open()` method loads the specified resource into the new or existing browsing context (window, `<iframe>` or tab) with the specified name. If the name doesn't exist, then a new browsing context is opened in a new tab or a new window, and the specified resource is loaded into it.

- **public static final**

A public static final variable is a compile-time constant, but a public final is just a final variable, i.e. you cannot reassign value to it but it's not a compile-time constant. This may look puzzling, but the actual difference allows how the compiler treats those two variables.

- **private RecyclerView**

`RecyclerView` makes it easy to efficiently display large sets of data. You supply the data and define how each item looks, and the `RecyclerView` library dynamically creates the elements when they're needed. As the name implies, `RecyclerView` recycles those individual elements. When an item scrolls off the screen, `RecyclerView` doesn't destroy its view. Instead, `RecyclerView` reuses the view for new items that have scrolled onscreen. This reuse vastly improves performance, improving your app's responsiveness and reducing power consumption.

- **private ArrayList**

An `ArrayList` class is a resizable array, which is present in the `java.util` package. While built-in arrays have a fixed size, `ArrayLists` can change their size dynamically. Elements can be added and removed from an `ArrayList` whenever there is a need, helping the user with memory management.

- **private SwipeRefreshLayout .**

In Android, the common "pull to refresh" UX concept is not built in to a ListView/RecyclerView. However, many Android applications would like to make use of this concept for their feeds. This is useful for all sorts of feeds such as a Twitter timeline. This effect can be achieved using the SwipeRefreshLayout class

- **private EditText**

A user interface element for entering and modifying text. When you define an edit text widget, you must specify the `android.R.styleable#TextView_inputType` attribute. For example, for plain text input set `inputType` to "text"

- **et_search.addTextChangedListener(new TextWatcher())**

Android EditText is a subclass of TextView. EditText is used for entering and modifying text. While using EditText widget, we must specify its input type in `inputType` property of EditText which configures the keyboard according to input. EditText uses `TextWatcher` interface to watch change made over EditText. For doing this, EditText calls the `addTextChangedListener()` method.

- **public abstract void afterTextChanged (Editable s)**

This method is called to notify you that, somewhere within `s`, the text has been changed. It is legitimate to make further changes to `s` from this callback, but be careful not to get yourself into an infinite loop, because any changes you make will cause this method to be called again recursively. (You are not told where the change took place because other `afterTextChanged()` methods may already have made other changes and invalidated the offsets. But if you need to know here, you can use `Spannable#setSpan` in `onTextChanged(CharSequence, int, int, int)` to mark your place and then look up from there where the span ended up.

- **Void beforeTextChanged**

This method is called to notify you that, within s, the count characters beginning at start are about to be replaced by new text with length after. It is an error to attempt to make changes to s from this callback.

- **void onTextChanged**

This method is called to notify you that, within s, the count characters beginning at start have just replaced old text that had length before. It is an error to attempt to make changes to s from this callback.

- **Others**

Void Filter, Void Fetchcountrywisedata(), Activity.showdialog(This), Void Onresponse, Void Onerrorresponse, Void Init(), Getsupportactionbar().settitle(), Getsupportactionbar().setDisplayhomeasupenabled(True), Getsupportactionbar().setDisplayshowhomeenabled(True), Void Onrefresh(), Void Loadcountrydata(), Piechart.startanimation(), Activity.dismissdialog(), Void Getintent(), Rivate String Version, Private Firebasedatabase, and More

CHAPTER 6

TESTING

6.1. INSTRUMENTED TESTING

Instrumented unit tests are tests that run on physical devices and emulators, and they can take advantage of the Android framework APIs and supporting APIs, such as AndroidX Test. Instrumented tests provide more fidelity than local unit tests, but they run much more slowly. Therefore, we recommend using instrumented unit tests only in cases where you must test against the behavior of a real device. AndroidX Test provides several libraries that make it easier to write instrumented unit tests when necessary. For example, Android Builder classes make it easier to create Android data objects that would otherwise be difficult to build.

To organize the execution of your instrumented unit tests, you can group a collection of test classes in a *test suite* class and run these tests together. Test suites can be nested; your test suite can group other test suites and run all their component test classes together.

A test suite is contained in a test package, similar to the main application package. By convention, the test suite package name usually ends with the `.suite` suffix (for example, `com.example.android.testing.mysample.suite`).

InstrumentedTest.java

```
import androidx.test.ext.junit.runners.AndroidJUnit4;
```

```
import org.junit.Test;
```

```
import org.junit.runner.RunWith;
```

```
import static org.junit.Assert.*;
```

```
/**
 * Instrumented test, which will execute on an Android device.
 *
 * @see <a href="http://d.android.com/tools/testing">Testing documentation</a>
 */
@RunWith(AndroidJUnit4.class)
public class ExampleInstrumentedTest {
    @Test
    public void useAppContext() {
        // Context of the app under test.
        Context appContext = InstrumentationRegistry.getInstrumentation().getTargetContext();

        assertEquals("com.codewithkalvin.covid19tracker", appContext.getPackageName());
    }
}
```

6.2. UNIT TESTING

You can evaluate your app's logic using local unit tests when you need to run tests more quickly and don't need the fidelity and confidence associated with running tests on a real device. With this approach, you normally fulfill your dependency relationships using either Robolectric or a mocking framework, such as Mockito. Usually, the types of dependencies associated with your tests determine which tool you use:

- If you have dependencies on the Android framework, particularly those that create complex interactions with the framework, it's better to include framework dependencies using Robolectric.
- If your tests have minimal dependencies on the Android framework, or if the tests depend only on your own objects, it's fine to include mock dependencies using a mocking framework like Mockito

UnitTest.java

```
package com.codewithkalvin.covid19tracker;

import org.junit.Test;

import static org.junit.Assert.*;

/**
 * Example local unit test, which will execute on the development machine (host).
 *
 * @see <a href="http://d.android.com/tools/testing">Testing documentation</a>
 */
public class ExampleUnitTest {
    @Test
    public void addition_isCorrect() {
        assertEquals(4, 2 + 2);
    }
}
```

CHAPTER 7

RESULTS

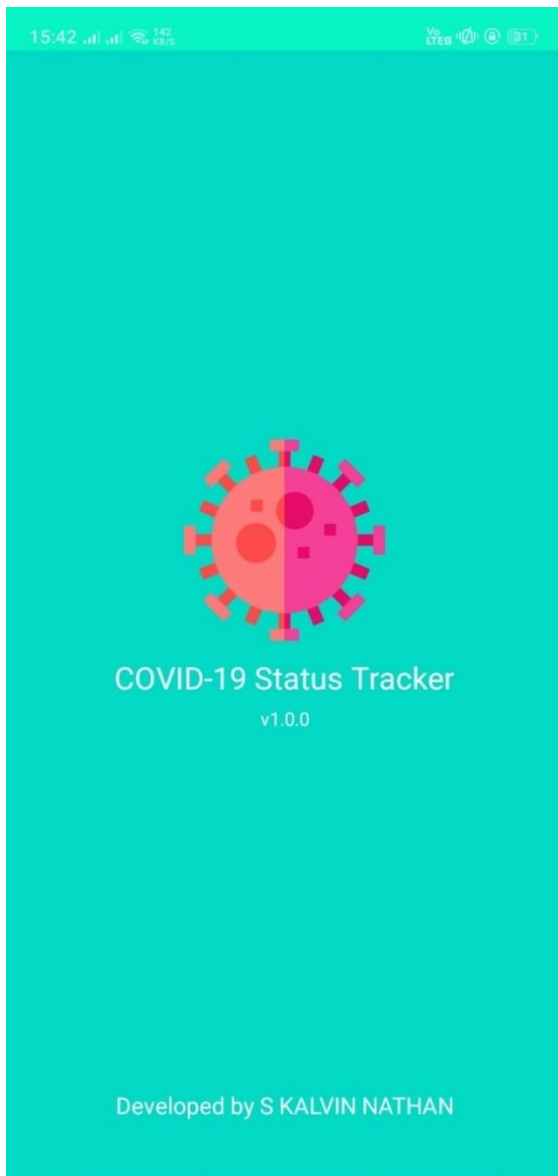


Figure 15: Front Page of App

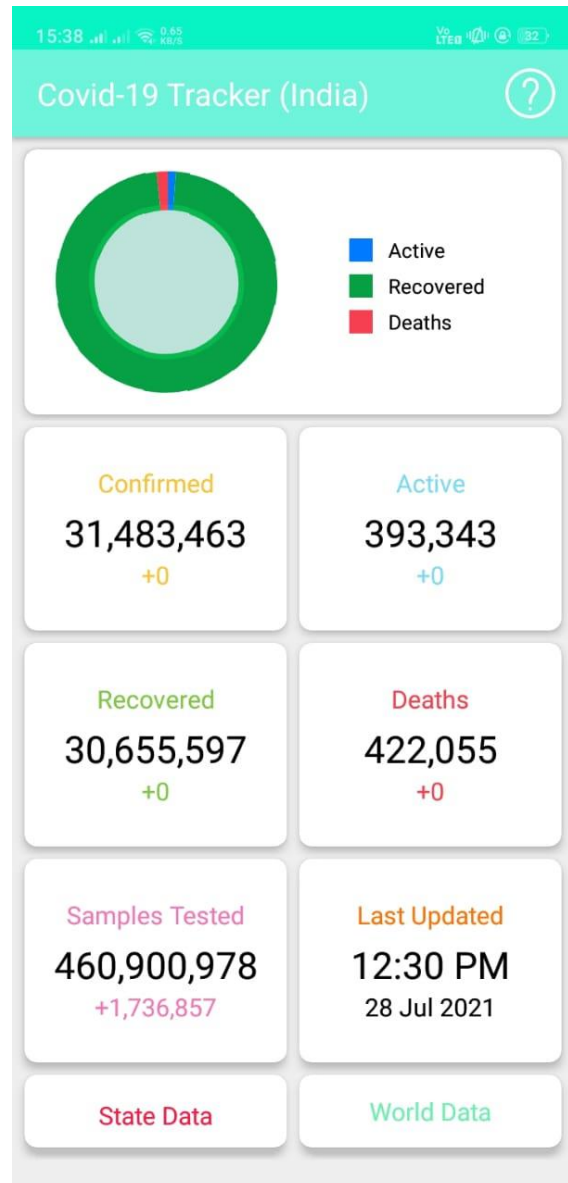


Figure 16: Home Page of App

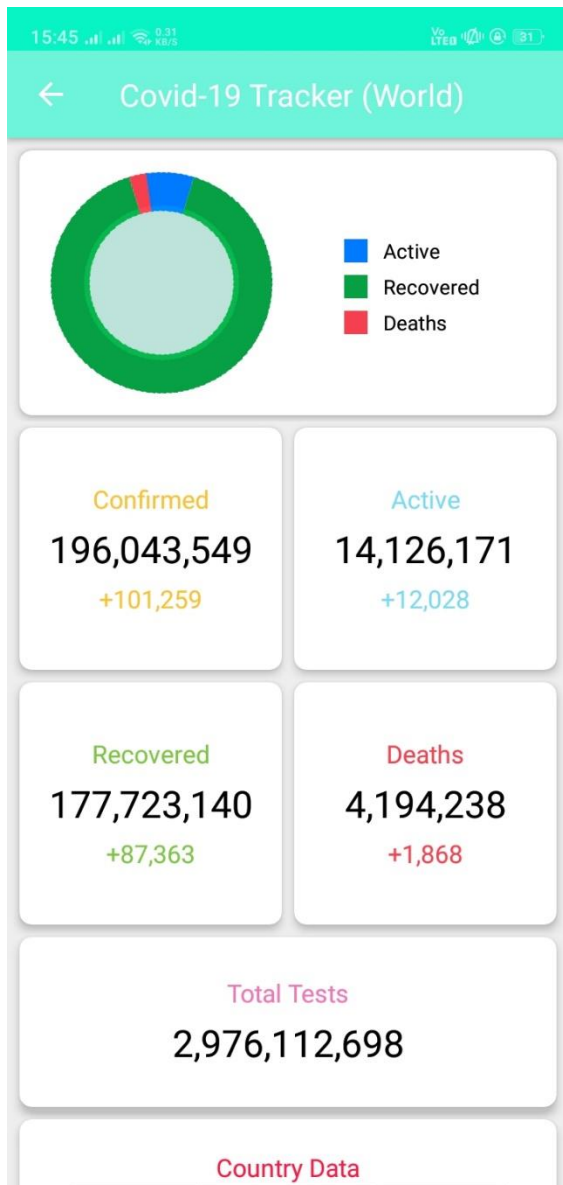


Figure 17: Active, Recovered, Death cases in world

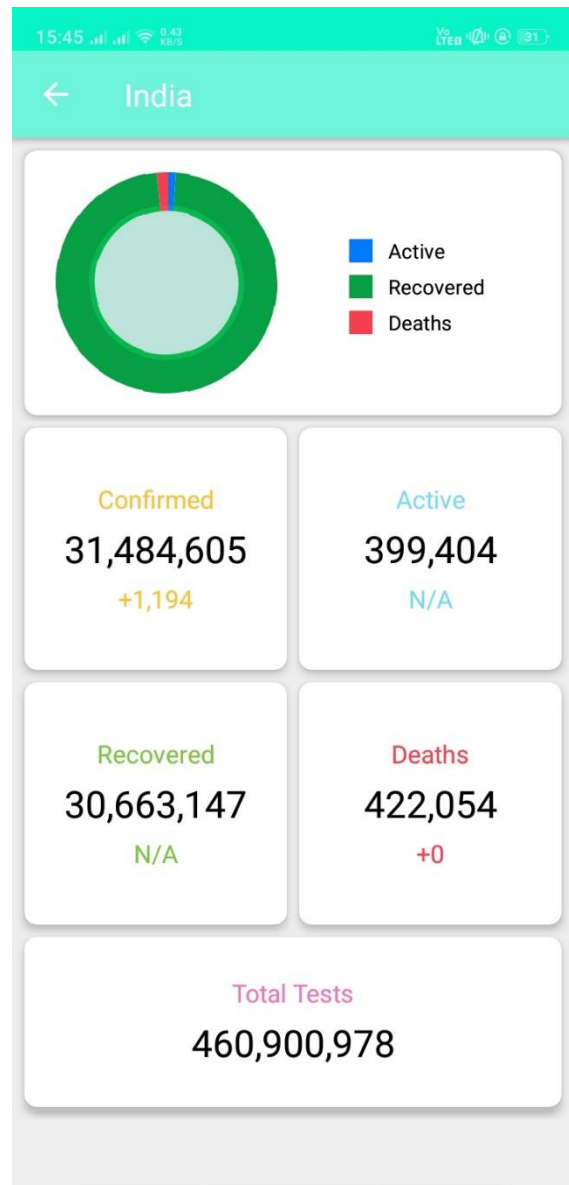


Figure 18: Active, Recovered, Death cases in India

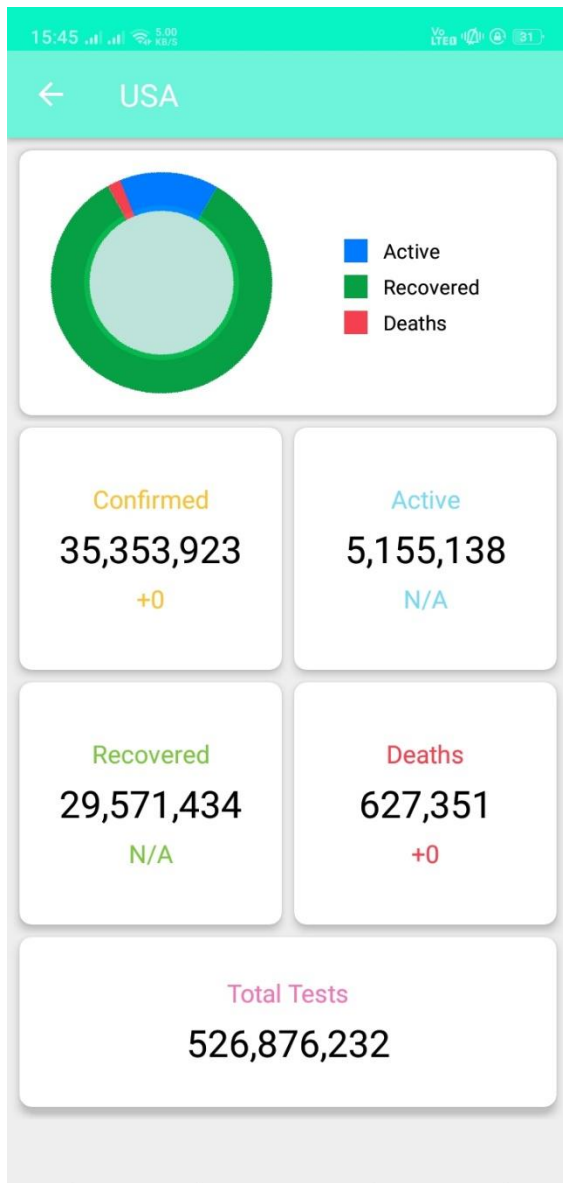


Figure 19: Active, Recovered, Death cases in USA

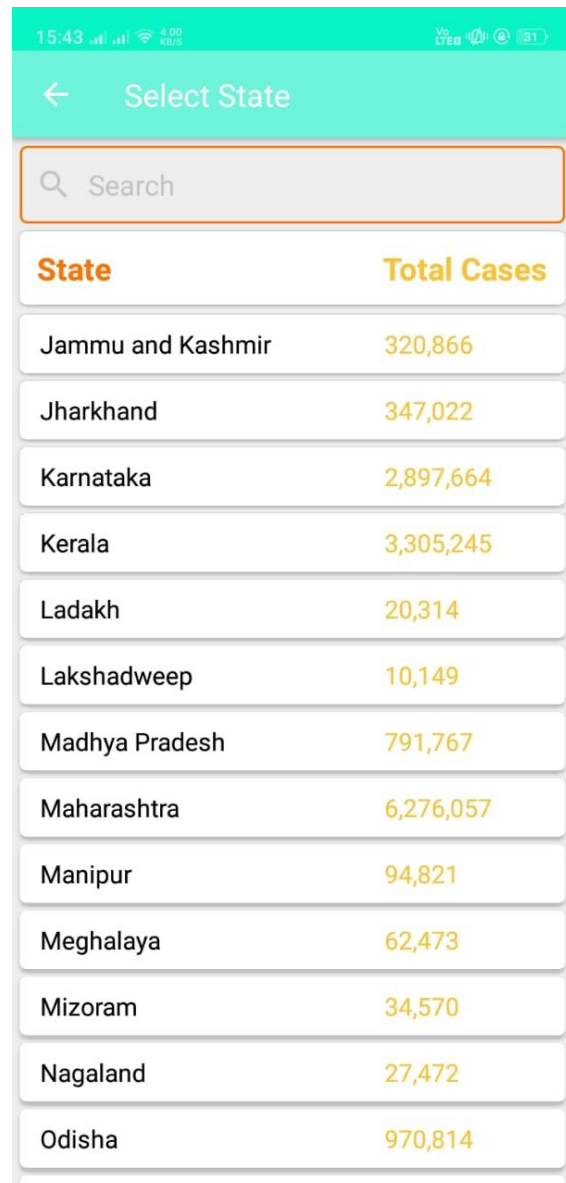
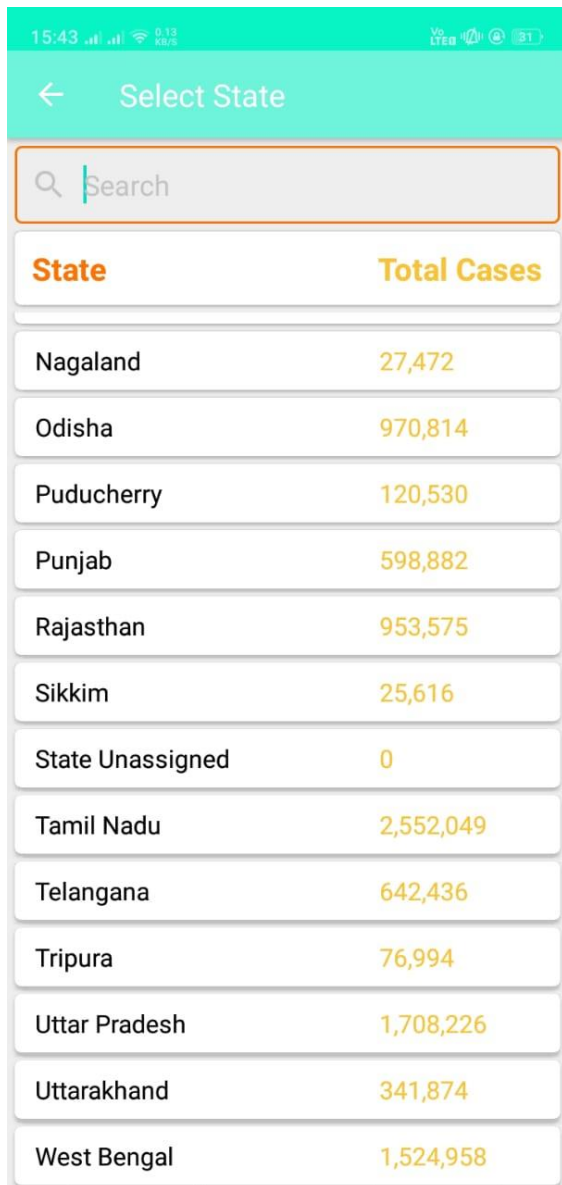
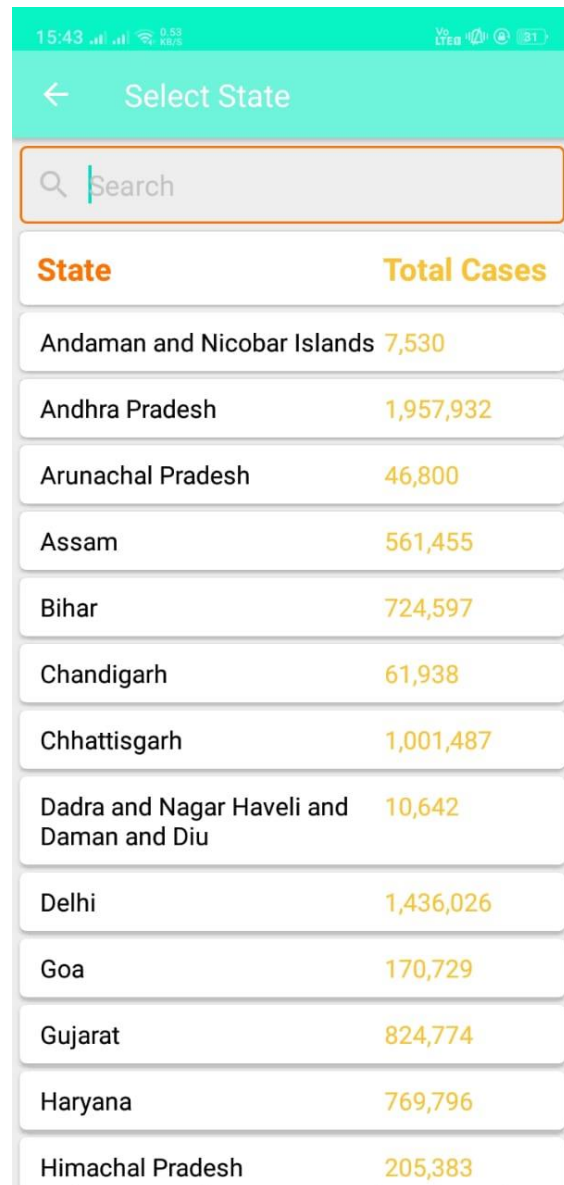


Figure 20: No of Total Cases In Each State



State	Total Cases
Nagaland	27,472
Odisha	970,814
Puducherry	120,530
Punjab	598,882
Rajasthan	953,575
Sikkim	25,616
State Unassigned	0
Tamil Nadu	2,552,049
Telangana	642,436
Tripura	76,994
Uttar Pradesh	1,708,226
Uttarakhand	341,874
West Bengal	1,524,958

Figure 20: No of Total Cases In Each State



State	Total Cases
Andaman and Nicobar Islands	7,530
Andhra Pradesh	1,957,932
Arunachal Pradesh	46,800
Assam	561,455
Bihar	724,597
Chandigarh	61,938
Chhattisgarh	1,001,487
Dadra and Nagar Haveli and Daman and Diu	10,642
Delhi	1,436,026
Goa	170,729
Gujarat	824,774
Haryana	769,796
Himachal Pradesh	205,383

Figure 20: No of Total Cases In Each State

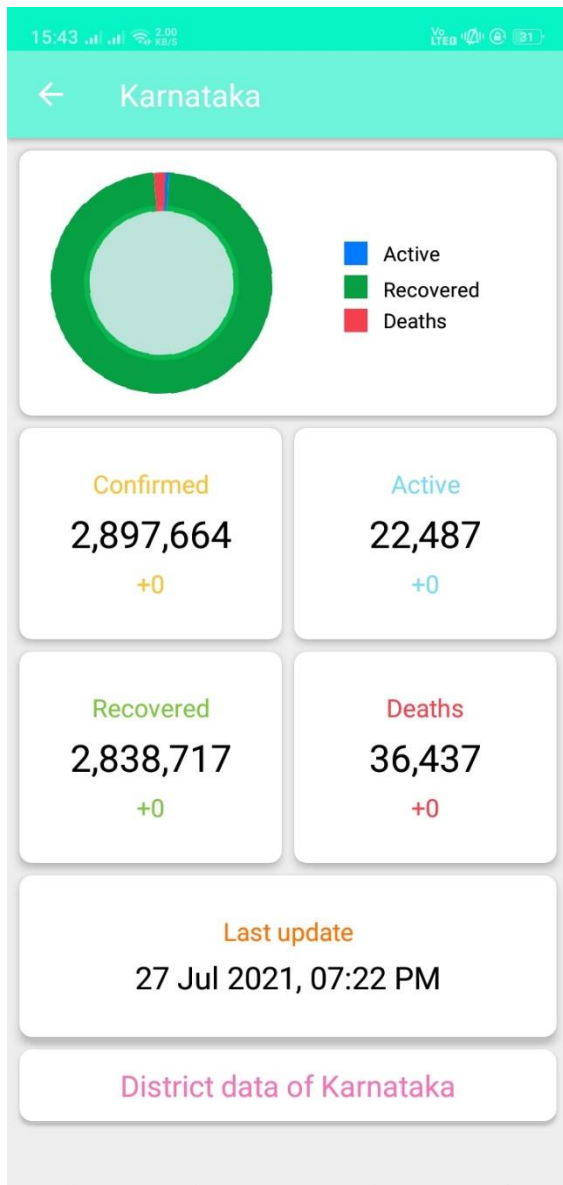


Figure 21: No of Total Cases In Karnataka State

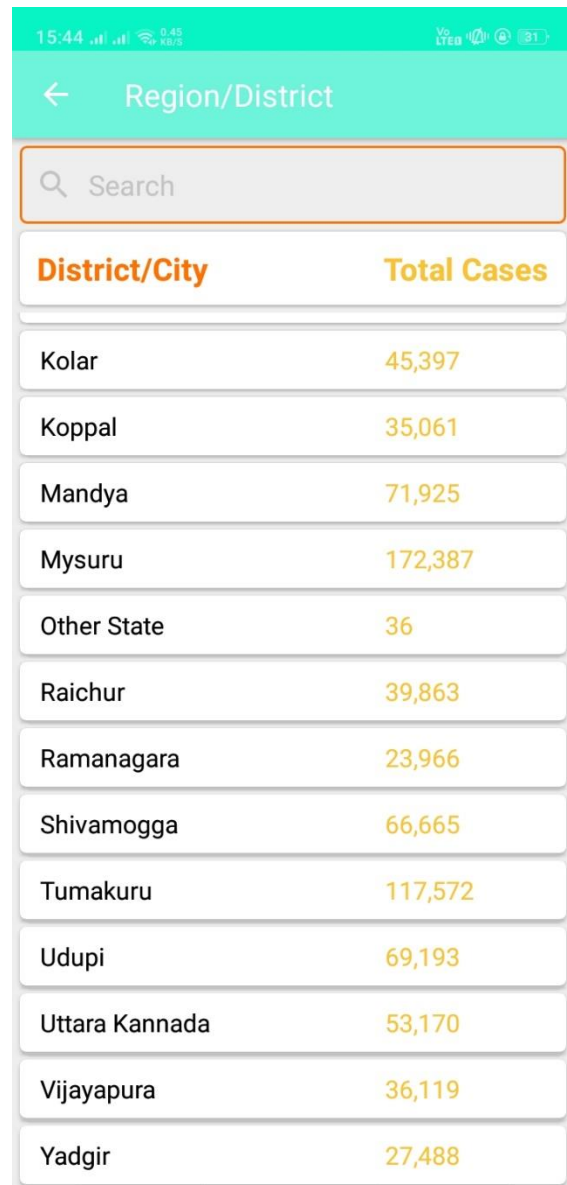
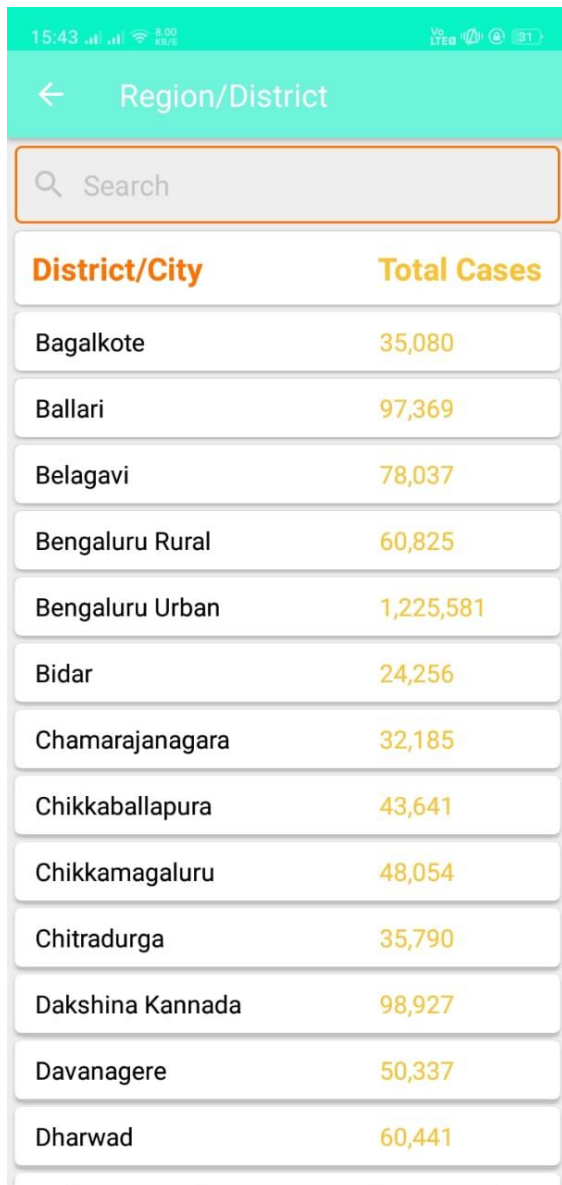


Figure 22: No of Total Cases In Each State



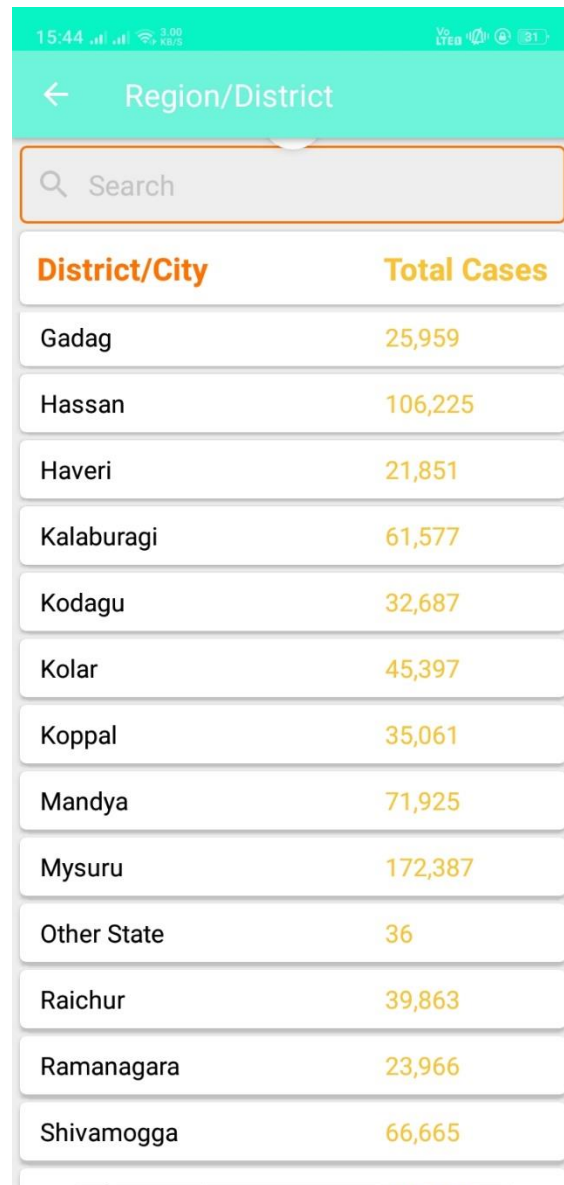
15:43 5.00 KB/S VoLTE

← Region/District

Search

District/City	Total Cases
Bagalkote	35,080
Ballari	97,369
Belagavi	78,037
Bengaluru Rural	60,825
Bengaluru Urban	1,225,581
Bidar	24,256
Chamarajanagara	32,185
Chikkaballapura	43,641
Chikkamagaluru	48,054
Chitradurga	35,790
Dakshina Kannada	98,927
Davanagere	50,337
Dharwad	60,441

Figure 22: No of Total Cases In Each State



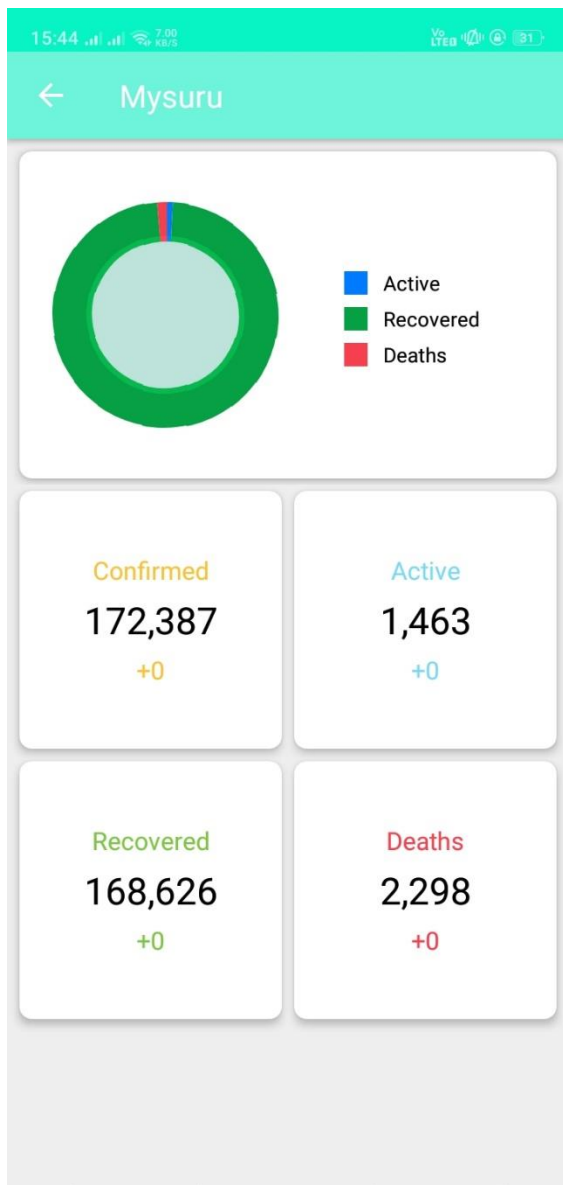
15:44 5.00 KB/S VoLTE

← Region/District

Search

District/City	Total Cases
Gadag	25,959
Hassan	106,225
Haveri	21,851
Kalaburagi	61,577
Kodagu	32,687
Kolar	45,397
Koppal	35,061
Mandya	71,925
Mysuru	172,387
Other State	36
Raichur	39,863
Ramanagara	23,966
Shivamogga	66,665

Figure 22: No of Total Cases In Each State



**Figure 23: No of Total Cases In
Bengaluru**



**Figure 24: No of Total Cases In
Mysuru**

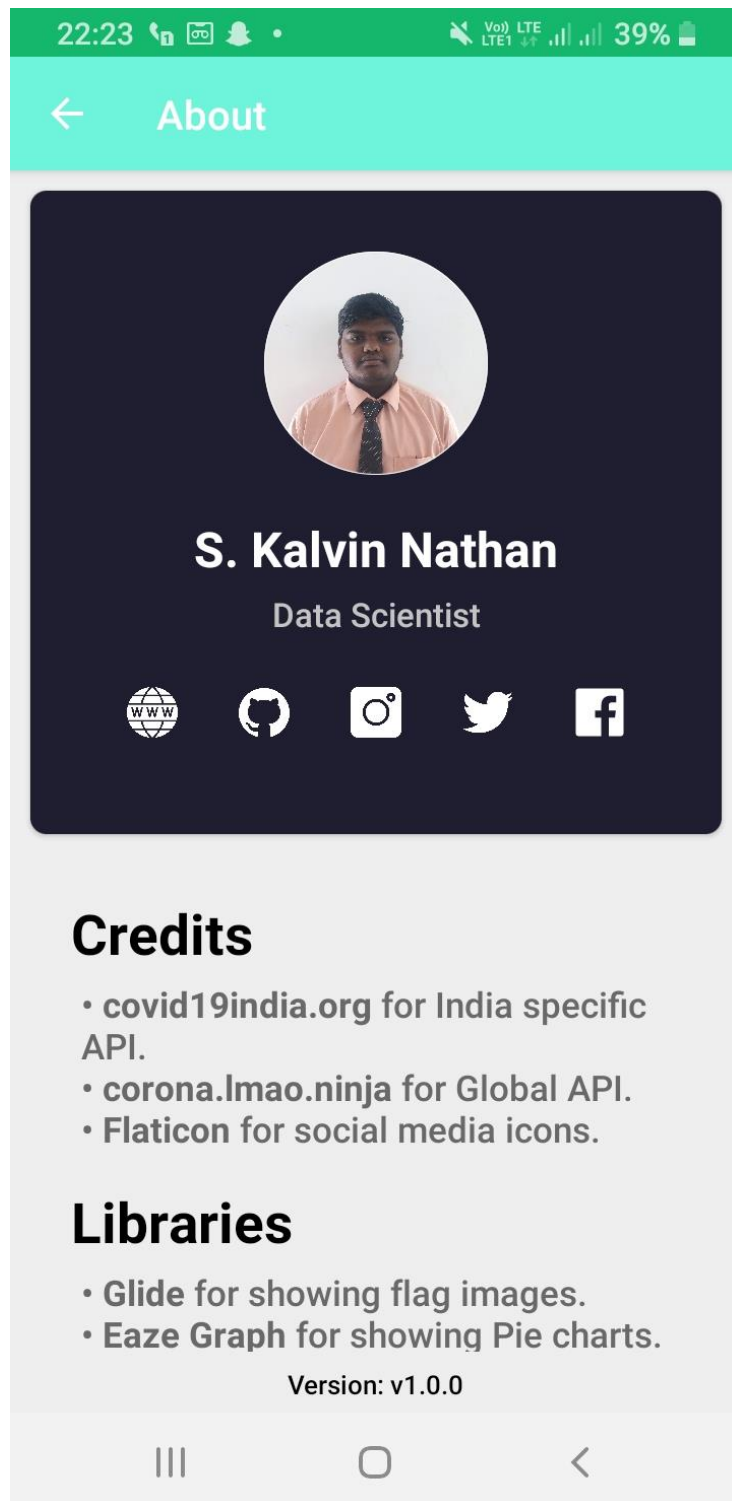


Figure 25: About Developer

CHAPTER 8

CONCLUSION

The application, overall, helps the government in keeping track of people who have been tested positive for the virus. It is also an excellent way to alert people about the number of infected cases in their area that have been identified as coronavirus-positive or if they accidentally came in contact with a person suffering from COVID-19. The application requires being in running mode at all times to continue tracing individuals actively. The API of the application can be used in such a way that it enables your smartphone to exchange the tracing keys periodically. In addition to showing the data of the number of users who have taken the self-assessment test and who have been identified as positive, a map can be shown of the nearby area where people have been identified as positive for COVID-19.

CHAPTER 9

REFERENCES

- World Health Organization. 2021. Coronavirus disease 2019 (COVID-19): Situation report. Retrieved from https://www.worldometers.info/coronavirus/?utm_campaign=homeAdvegas1?%20
- The Government of India. 2020. Live updates from Worldometer. Retrieved from <https://www.worldometers.info/coronavirus/country/india/>.
- **covid19india.org** for India specific API.
- **.corona.Imao.ninja** for Global API Flaticon for social media icons.