

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**  
**“Jnana Sangama”, Belagavi-590018, Karnataka**



**BANGALORE INSTITUTE OF TECHNOLOGY**  
**K. R. Road, V. V. Puram, Bengaluru-560 004**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**MOBILE APPLICATION DEVELOPMENT**  
**MINI PROJECT (18CSMP68)**

**“Covid-19-Tracker”**

**Submitted By**

**1BI18CS134**

**SANJEEV KUMAR MENDEGAR**

**for the academic year 2020-21**

**Under the guidance of**

**Prof. Kanchan Ambarish Purohit**  
Assistant Professor

**Prof. M S Bhargavi**  
Assistant Professor

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**  
**“Jnana Sangama”, Belagavi-590018, Karnataka**

**BANGALORE INSTITUTE OF TECHNOLOGY**  
**K. R. Road, V. V. Puram, Bengaluru-560 004**



**Department of Computer Science & Engineering**

***Certificate***

This is to certify that the implementation of **Mobile Application Development MINI PROJECT (18CSMP68)** entitled “**Covid-19-Tracker**” has been successfully completed by **1B18CS134** **SANJEEV KUMAR MENDEGAR**

of VI semester B.E. for the partial fulfillment of the requirements for the Bachelor's degree in **Computer Science & Engineering** of the **Visvesvaraya Technological University** during the academic year **2020-2021**.

**Lab In charges:**

**Prof. Kanchan Ambarish**  
**Purohit**

Assistant Professor  
Dept. of CS&E, BIT

**Prof. M S Bhargavi**

Assistant Professor  
Dept. of CS&E, BIT

**Dr. Asha T.**

Professor and Head  
Department of CS&E  
Bangalore Institute of Technology

Examiners: 1)

2)

## ACKNOWLEDGEMENT

The knowledge & satisfaction that accompany the successful completion of any task would be incomplete without mention of people who made it possible, whose guidance and encouragement crowned my effort with success. I would like to thank all and acknowledge the help I have received to carry out this Mini Project.

I would like to convey my thanks to Head of Department **Dr. ASHA T.** for being kind enough to provide the necessary support to carry out the mini project. I am most humbled to mention the enthusiastic influence provided by the lab in-charges **Prof. Kanchan Ambarish Purohit** and **Prof. M S Bhargavi**, on the project for their ideas, time to time suggestions for being a constant guide and co-operation showed during the venture and making this project a great success.

I would also take this opportunity to thank my friends and family for their constant support and help. I'm very much pleased to express my sincere gratitude to the friendly co-operation showed by all the **staff members** of Computer Science Department, BIT.

**SANJEEV KUMAR MENDEGAR**  
**1BI18CS134**

## Table of contents

1. Introduction.....	1
1.1 Project Summary.....	1
1.2 Project Purpose.....	2
1.3 Project Scope.....	2
1.4 Technology and Literature Review.....	2
1.5 Problem Statement.....	2
1.6 Objective Of The Project .....	3
1.7 Organization Of The Report.....	3
2. System Requirements .....	4
2.1 User Characteristics.....	4
2.2 Hardware and Software Requirements.....	4
2.3 Project Operation Constraints.....	5
3. Design.....	6
3.1 System Architecture Design.....	6
3.2 Data Flow Graph.....	7
3.3 Control Flow Graph.....	8
3.4 State Transition Diagram.....	9
3.5. Module Description.....	9
4. Implementation.....	10
4.1 Built In Functions.....	10
4.2 Android Manifest and Special Permissions.....	12
4.3 XML and Java Code.....	13
5. Snapshots.....	29

6. Conclusion.....	32
Future Enhancement.....	32
Bibliography.....	33

# Chapter -1

## INTRODUCTION

### 1.1 Project Summary

The World Health Organization has declared the outbreak of the novel coronavirus, Covid-19 as pandemic across the world. With its alarming surge of affected cases throughout the world, lockdown, and awareness (social distancing, use of masks etc.) among people are found to be the only means for restricting the community transmission. In a densely populated country like India, it is very difficult to prevent the community transmission even during lockdown without social awareness and precautionary measures taken by the people. Recently, several containment zones had been identified throughout the country and divided into red, orange and green zones, respectively. The red zones indicate the infection hotspots, orange zones denote some infection and green zones indicate an area with no infection. This paper mainly focuses on development of an Android application which can inform people of the Covid-19 containment zones and prevent trespassing into these zones.

### 1.2 Project Purpose

The objective of this app was to review the functionalities and effectiveness of the free mobile health applications available in github.

We have developed a mobile based application to provide information regarding the Covid-19 cases in the world. The application further tracks based on country and provides stats of covid 19 cases, if the user has entered a specific country. The application also provides daily Covid-19 cases statistics to the users to keep them updated. The application is developed on Android SDK. The application also uses RESTful web services to show the Covid-19 cases around the world.

### 1.3 Project Scope

The world is facing one of the worst epidemics, the outbreak of COVID-19, you all are aware of that. So during this lockdown time . so we have created a **COVID-19 Tracker Android App using REST API** which will track the **Global Stats and filtered on the basis of Countries too.**

### 1.4 Technology and Literature Review

In the recent years, the advances in mobile technology have brought an exorbitant change in daily lifestyle of individuals. Smartphones/mobile devices are rampant in all aspects of human life. This has led to an extreme demand for developing software that runs on mobile devices. The developers have to keep up with this high demand and deliver high-quality app on time and within budget. For this, estimation of development and testing of apps play a pivotal role. In this paper, a Systematic Literature Review (SLR) is conducted to highlight development and testing estimation process for software/application. The goal of the present literature survey is to identify and compare existing test estimation techniques for traditional software (desktop/laptop) and for mobile software/application. The characteristics that make mobile software/application different from traditional software are identified in this literature survey. Further, the trend for developing the software is towards agile, thus this study also presents and compares estimation techniques used in agile software development for mobile applications. The analysis of literature review suggests filling a research gap to present formal models for estimating mobile application considering specific characteristics of mobile software.

### 1.5 Problem Statement

COVID-19! is a android-based Corona tracker mobile app. This android application acts as a portal for the public to keep track of COVID-19. All the contents are handpicked and filtered to the best extent to ensure that sources are reliable with minimal hoaxes, in the best interest of the public.

## **1.6 Objective Of The Project**

The survey shows that there are several apps developed in the country to fight and contain COVID-19. Most of the states of our country have their own apps with specific features and functionality to help their citizens to stop COVID-19 spread, get medical assistance during a crisis, create awareness, and understand safety precautions. The study also shows that there are a limited number of apps which show the COVID-19 containment zones in the country or state and out of these none has the functionality of notifying and alerting the user when they have entered a containment zone. Therefore, no app in the Google Playstore is comparable with our proposed application because the idea behind the development of the proposed app is different. This highlights the novelty of the proposed app.

## **1.7 Organization Of The Report**

The project was organised in a systematic way. First we analysed what are the basic features to be included in the project to make it acceptable. As it is a android project, we made the blueprint of xml design prior (Include it in the report), so as to have an idea like how our output must look like. After all these, the source code was formulated as a paper work. All the required software were downloaded. Finally, the successful implementation of the project.



# Chapter -2

## SYSTEM SPECIFICATION

### 2.1 User Characteristics

- User-friendly
- Simple interface.
- Speed-Fast loading screens.
- Most Convenient
- Flexibility
- Search Options
- Good Image resolution
- Security
- Bright and bold colour schemes
- Dark mode adaptable
- Notification on Internet disable

### 2.2 Hardware and Software Requirements

- A 64-bit environment is required for Android 2.3.x (Gingerbread) and higher versions, including the master branch. You can compile older versions on 32-bit systems.
- At least 250GB of free disk space to check out the code and an extra 150 GB to build it. If you conduct multiple builds, you need additional space.  
**Note:** If you're checking out a mirror, you need more space as full Android Open Source Project (AOSP) mirrors contain all Git repositories that have ever been used.
- At least 4 GB of available RAM is required.
- Microsoft Windows 7/8/10 (32 or 64 bit).
- 2GB RAM.
- 4GB RAM recommended.
- 500 MB disk space
- 1 GB for Android SDK.
- Java Development Kit (JDK) 7.
- 1280x800 screen resolution.
- A faster processor (according to your budget).

## **2.3 Project Operation Constraints**

### **1. Quality**

Quality is one of six major constraints of every project.

### **2. Time**

One of the most important stakeholder considerations, project time (how long it will take to deliver), is a vital measure of project success. Your task is to estimate project time as accurately as possible, which requires a blend of research and experience.

### **3.Scope**

Since a project scope is not an estimate but a guaranteed set of deliverables, it's difficult to imagine creating a range for this project constraint.

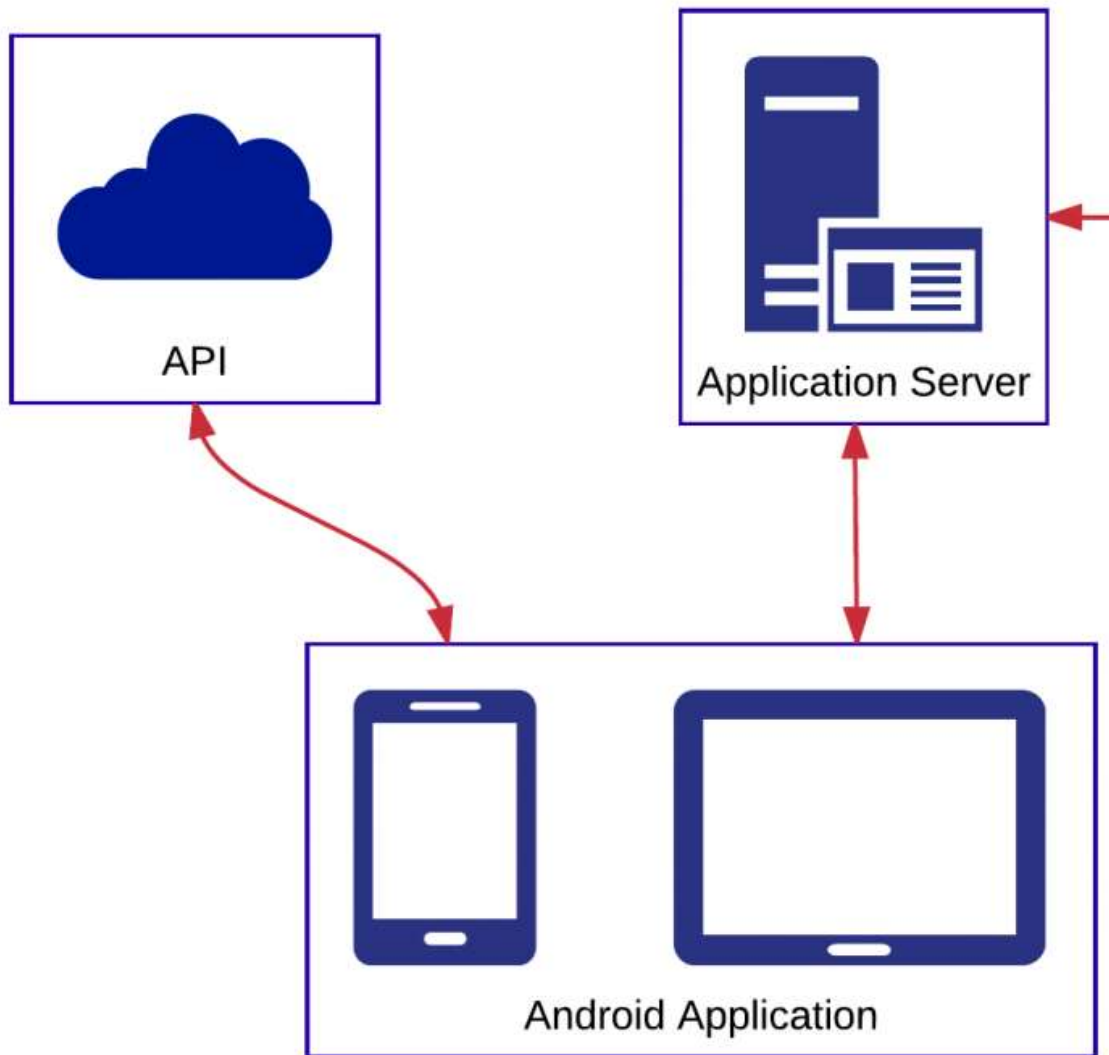
### **4.Benefits**

The projected benefits of any project should be clearly think before project planning. To put it simply, a project's value must be determined early and fully agreed upon before launch.

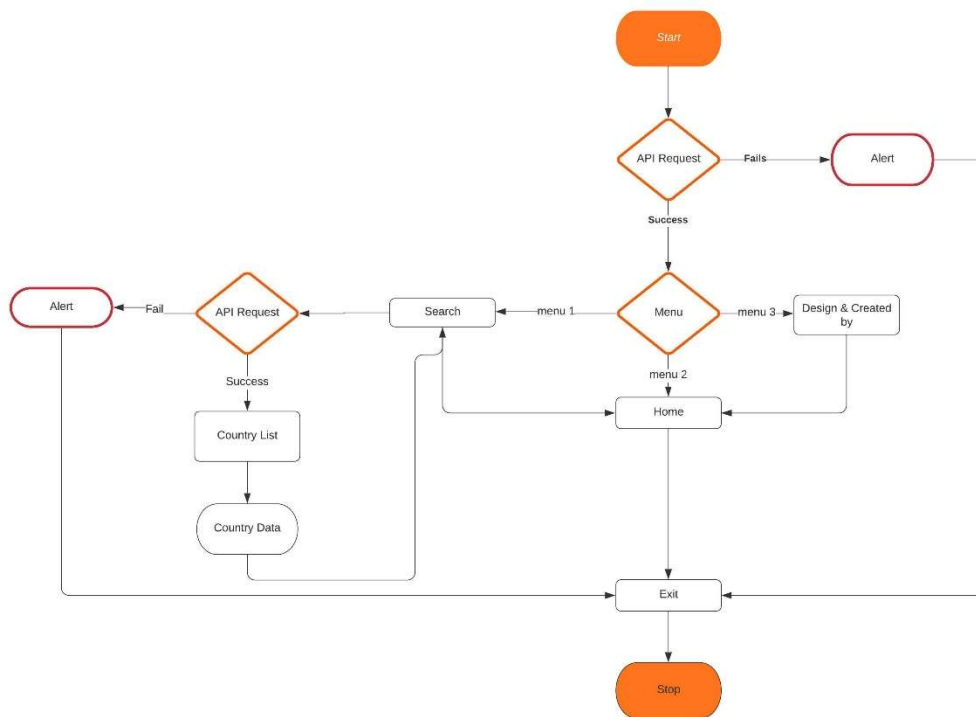
## Chapter -3

### DESIGN

#### 3.1 System Architecture Design

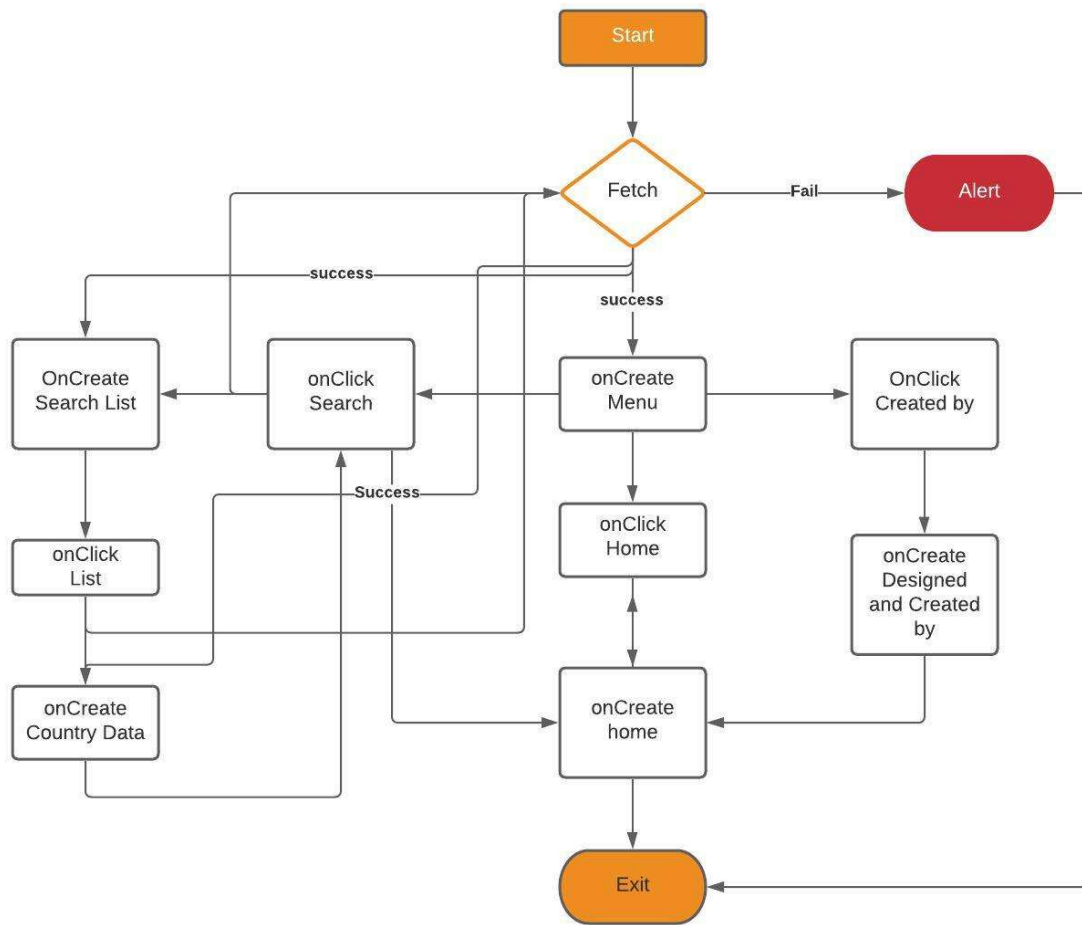


### 3.2 Data Flow Graph



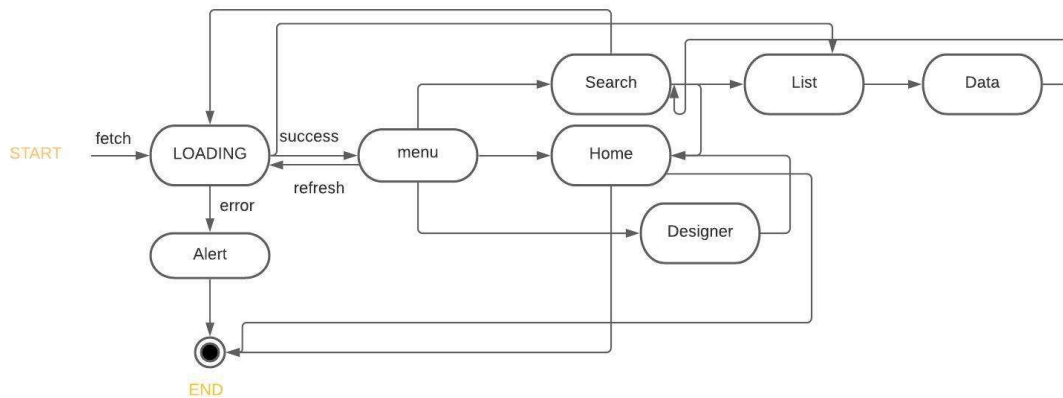
*Figure 3.2 Data Flow Graph*

### 3.3 Control Flow Graph



*Figure 3.3 Control Flow Graph*

### 3.4 State Transition Diagram



*Figure 3.4 State Transition Diagram*

### 3.5 Module Description

The program consists of following modules:

- **Home:** In the Home activity basically global Stats of covid-19 cases and pie chart is shown.
- **Search:** In this activity the country lists are shown.
- **Country Data:** Country wise data of covid-19 cases are displayed with their pie chart
- **Designed & Created By:** Here details of project creators are shown.

# Chapter -4

## IMPLEMENTATION

### 4.1 Built in Functions

#### 1. Relative Layout:

RelativeLayout is a view group that displays child views in relative positions. The position of each view can be specified as relative to sibling elements or in positions relative to the parent RelativeLayout area.

#### 2.Card View:

Apps often need to display data in similarly styled containers. These containers are often used in lists to hold each item's information. The system provides the CardView API as an easy way for you to show information inside cards that have a consistent look across the platform. These cards have a default elevation above their containing view group, so the system draws shadows below them. Cards provide an easy way to contain a group of views while providing a consistent style for the container.

#### 3.Linear Layout:

LinearLayout is a view group that aligns all children in a single direction, vertically or horizontally. You can specify the layout direction with the android:orientation attribute.

#### 4.Pie Chart:

A **pie chart** is a circular statistical graphic, which is divided into slices to illustrate numerical proportions. It depicts a special chart that uses “pie slices”, where each sector shows the relative sizes of data. A circular chart cuts in a form of radii into segments describing relative frequencies or magnitude also known as circle graph. A pie chart represents numbers in percentages, and the total sum of all segments needs to equal 100%.

**5.Loader:**

The Loader API lets you load data from a content provider or other data source for display in an `FragmentActivity` or `Fragment`.

**6.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.

**7.Frame Layout:**

`FrameLayout` is designed to block out an area on the screen to display a single item.

**8.Meow Bottom Navigation:** A simple & curved & material bottom navigation for Android.

**9.Dialog box:**

A dialog is a small window that prompts the user to make a decision or enter additional information.

**10. setOnClickMenuListener():**

To make click event work add `android:onClick` attribute to the `Button` element in your XML layout. The value for this attribute must be the name of the method you want to call in response to a click event. The Activity hosting the layout must then implement the corresponding method.

**11. StringRequest():**

Android Volley `StringRequest`. `StringRequest` is used when you want the response returned in the form of a `String`.

**12.Volley Library():**

Volley is an HTTP library that makes networking for Android apps easier and most importantly, faster.

**13. getSupportActionBar():**

To use the `ActionBar` utility methods, call the activity's `getSupportActionBar()` method. This method returns a reference to an `appcompat ActionBar` object.



**14.Adapter:**

An Adapter object acts as a bridge between an AdapterView and the underlying data for that view. The Adapter provides access to the data items. The Adapter is also responsible for making a View for each item in the data set.

**15.Intent:**

An Intent is a messaging object you can use to request an action from another app component.

**16.Glide:**

Glide is a fast and efficient open source media management and image loading framework for Android that wraps media decoding, memory and disk caching, and resource pooling into a simple and easy to use interface.

**17.REST Api:**

A REST API defines a set of functions which developers can perform requests and receive responses via HTTP protocol such as GET and POST.

## **4.2 Android Manifest and Special Permissions**

**1.Reading network state:**

- ConnectivityManager tells your app about the state of connectivity in the system.
- The Network class represents one of the networks that the device is currently connected to. You can use the Network object as a key to gather information about the network with ConnectivityManager or to bind sockets on the network. When the network disconnects, the Network object stops being usable; even if the device later reconnects to the same appliance, a new Network object will represent the new network.

**2.Connect to the network:**

To perform network operations in our application.

**3.Dark Theme:**

- Dark theme is available in Android 10 (API level 29) and higher. It has many benefits:
- Can reduce power usage by a significant amount (depending on the device's screen technology).
- Improves visibility for users with low vision and those who are sensitive to bright light.
- Makes it easier for anyone to use a device in a low-light environment.

### 4.3 XML and Java Code

#### Activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/white"
    tools:context=".MainActivity">

    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="180dp"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
        android:layout_marginTop="20dp"
        app:cardElevation="10dp"
        android:id="@+id/cardViewGraph"
        app:cardCornerRadius="10dp"
        android:outlineSpotShadowColor="@color/deaths">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="horizontal"
            android:weightSum="2"
            >
            <org.eazegraph.lib.charts.PieChart
                android:id="@+id/piechart"
                android:layout_width="0dp"
                android:layout_height="match_parent"
                android:padding="6dp"
                android:layout_weight="1"
                android:layout_marginTop="15dp"
                android:layout_marginLeft="15dp"
                android:layout_marginBottom="15dp"
            />
            <LinearLayout
                android:layout_width="0dp"
                android:layout_height="match_parent"
                android:layout_weight="1"
```

```
    android:layout_marginLeft="20dp"
    android:orientation="vertical"
    android:gravity="center_vertical">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="15dp"
        android:gravity="center_vertical">
        <View
            android:layout_width="15dp"
            android:layout_height="match_parent"
            android:background="@color/cases"/>
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Total Cases"
            android:paddingLeft="10dp"/>

    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="15dp"
        android:gravity="center_vertical"
        android:layout_marginTop="5dp">
        <View
            android:layout_width="15dp"
            android:layout_height="match_parent"
            android:background="@color/recovered"/>
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Recovered"
            android:paddingLeft="10dp"/>

    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="15dp"
        android:gravity="center_vertical"
        android:layout_marginTop="5dp">
        <View
            android:layout_width="15dp"
            android:layout_height="match_parent"
            android:background="@color/deaths"/>
```

---

```

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Deaths"
            android:paddingLeft="10dp"/>

    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="15dp"
        android:gravity="center_vertical"
        android:layout_marginTop="5dp">
        <View
            android:layout_width="15dp"
            android:layout_height="match_parent"
            android:background="@color/active"/>
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Active"
            android:paddingLeft="10dp"/>

    </LinearLayout>
</LinearLayout>

</LinearLayout>

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="330dp"
    android:layout_below="@+id/cardViewGraph"
    android:outlineSpotShadowColor="@color/active"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginTop="20dp"
    app:cardElevation="10dp"
    app:cardCornerRadius="10dp"
    >
<!--      android:layout_above="@+id/btnTrack"-->
<RelativeLayout

```

---

```
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    >
    <com.leo.simplearcloader.SimpleArcLoader
        android:visibility="visible"
        android:id="@+id/loader"
        android:layout_centerInParent="true"
        android:layout_width="60dp"
        android:layout_height="60dp"
        app:arc_style="simple_arc"
        >
    </com.leo.simplearcloader.SimpleArcLoader>

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/scrollStats"
        android:visibility="gone">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="global stats"
                android:textAllCaps="true"
                android:textSize="28sp"
                android:textStyle="bold"
                android:layout_marginLeft="25dp"
                android:layout_marginTop="20dp"/>

            <View
                android:layout_width="match_parent"
                android:layout_height="1dp"
                android:background="@color/color_two"
                android:layout_marginLeft="20dp"
                android:layout_marginRight="20dp"
                android:layout_marginTop="5dp"/>

            <RelativeLayout
                android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
        android:layout_marginRight="25dp"
        android:layout_marginLeft="25dp"
        android:layout_marginTop="10dp"
        android:layout_marginBottom="10dp">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:fontFamily="sans-serif-light"
            android:text="Cases"
            android:textSize="18sp"/>
        <TextView
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="0"
            android:id="@+id/tvCases"
            android:textAlignment="textEnd"
            android:textSize="18sp"

            android:textColor="@color/cases"
            android:textStyle="bold"
            android:fontFamily="sans-serif-light"
            android:layout_alignParentRight="true"/>

    </RelativeLayout>
    <View
        android:layout_width="match_parent"
        android:layout_height="1dp"
        android:background="@color/color_two"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
    />
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginRight="25dp"
        android:layout_marginLeft="25dp"
        android:layout_marginTop="10dp"
        android:layout_marginBottom="10dp">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
```

```
        android:fontFamily="sans-serif-light"
        android:text="Recovered"
        android:textSize="18sp"/>
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="0"
        android:id="@+id/tvRecovered"
        android:textAlignment="textEnd"
        android:textSize="18sp"
        android:textColor="@color/recovered"
        android:textStyle="bold"
        android:fontFamily="sans-serif-light"
        android:layout_alignParentRight="true"/>

</RelativeLayout>
<View
    android:layout_width="match_parent"
    android:layout_height="1dp"
    android:background="@color/color_two"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
/>
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginRight="25dp"
    android:layout_marginLeft="25dp"
    android:layout_marginTop="10dp"
    android:layout_marginBottom="10dp">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:fontFamily="sans-serif-light"
        android:text="Critical"
        android:textSize="18sp"/>
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="0"
        android:id="@+id/tvCritical"
        android:textAlignment="textEnd"
        android:textSize="18sp"
```

```
        android:textColor="@color/color_one"
        android:textStyle="bold"
        android:fontFamily="sans-serif-light"
        android:layout_alignParentRight="true"/>

</RelativeLayout>
<View
    android:layout_width="match_parent"
    android:layout_height="1dp"
    android:background="@color/color_two"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
/>
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginRight="25dp"
    android:layout_marginLeft="25dp"
    android:layout_marginTop="10dp"
    android:layout_marginBottom="10dp">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:fontFamily="sans-serif-light"
        android:text="Active"
        android:textSize="18sp"/>

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="0"
        android:id="@+id/tvActive"
        android:textAlignment="textEnd"
        android:textSize="18sp"

        android:textColor="@color/active"
        android:textStyle="bold"
        android:fontFamily="sans-serif-light"
        android:layout_alignParentRight="true"/>

</RelativeLayout>
<View
    android:layout_width="match_parent"
```



```
        android:layout_height="1dp"
        android:background="@color/color_two"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
    />
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginRight="25dp"
        android:layout_marginLeft="25dp"
        android:layout_marginTop="10dp"
        android:layout_marginBottom="10dp">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:fontFamily="sans-serif-light"
            android:text="Today Cases"
            android:textSize="18sp"/>
        <TextView
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="0"
            android:id="@+id/tvTodayCases"
            android:textAlignment="textEnd"
            android:textSize="18sp"

            android:textColor="@color/color_one"
            android:textStyle="bold"
            android:fontFamily="sans-serif-light"
            android:layout_alignParentRight="true"/>

    </RelativeLayout>
    <View
        android:layout_width="match_parent"
        android:layout_height="1dp"
        android:background="@color/color_two"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
    />
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginRight="25dp"
```

```
        android:layout_marginLeft="25dp"
        android:layout_marginTop="10dp"
        android:layout_marginBottom="10dp">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:fontFamily="sans-serif-light"
            android:text="Total Deaths"
            android:textSize="18sp"/>
        <TextView
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="0"
            android:id="@+id/tvTotalDeaths"
            android:textAlignment="textEnd"
            android:textSize="18sp"

            android:textColor="@color/deaths"
            android:textStyle="bold"
            android:fontFamily="sans-serif-light"
            android:layout_alignParentRight="true"/>

    </RelativeLayout>
    <View
        android:layout_width="match_parent"
        android:layout_height="1dp"
        android:background="@color/color_two"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
    />
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginRight="25dp"
        android:layout_marginLeft="25dp"
        android:layout_marginTop="10dp"
        android:layout_marginBottom="10dp">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:fontFamily="sans-serif-light"
            android:text="Today Deaths"
```

```
        android:textSize="18sp"/>
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="0"
        android:id="@+id/tvTodayDeaths"
        android:textAlignment="textEnd"
        android:textSize="18sp"

        android:textColor="@color/color_one"
        android:textStyle="bold"
        android:fontFamily="sans-serif-light"
        android:layout_alignParentRight="true"/>

</RelativeLayout>
<View
    android:layout_width="match_parent"
    android:layout_height="1dp"
    android:background="@color/color_two"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
/>
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginRight="25dp"
    android:layout_marginLeft="25dp"
    android:layout_marginTop="10dp"
    android:layout_marginBottom="10dp">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:fontFamily="sans-serif-light"
        android:text="Affected Countries"
        android:textSize="18sp"/>
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="0"
        android:id="@+id/tvAffectedCountries"
        android:textAlignment="textEnd"
        android:textSize="18sp"
        android:textColor="@color/color_one"
```

```
        android:textStyle="bold"
        android:fontFamily="sans-serif-light"
        android:layout_alignParentRight="true"/>

    </RelativeLayout>
    <View
        android:layout_width="match_parent"
        android:layout_height="1dp"
        android:background="@color/color_two"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
    />

    </LinearLayout>

    </ScrollView>

    </RelativeLayout>

</androidx.cardview.widget.CardView>

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="550dp"
    android:id="@+id/r"
    android:orientation="vertical"
    android:layout_alignParentBottom="true">

    <FrameLayout
        android:id="@+id/frame"
        android:layout_width="match_parent"
        android:layout_height="0dp"

        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

    <com.etebarian.meowbottomnavigation.MeowBottomNavigation
        android:id="@+id/bottom_navigation"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

```
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:mbn_backgroundBottomColor="@color/white"
app:mbn_circleColor="#fda102"
app:mbn_countBackgroundColor="#ff6f00"
app:mbn_countTextColor="#ffffff"
app:mbn_defaultIconColor="#90a4ae"
app:mbn_rippleColor="#2f424242"
app:mbn_selectedIconColor="#3c415e"
app:mbn_shadowColor="@color/active" />
```

```
</RelativeLayout>
```

```
</RelativeLayout>
```

MainActivity.java

```
package com.example.cov19trac;

import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;

import android.app.Dialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Color;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.Bundle;
import android.view.View;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.ScrollView;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
```

```
import com.etebarian.meowbottomnavigation.MeowBottomNavigation;
import com.leo.simplearcloader.SimpleArcLoader;

import org.eazegraph.lib.charts.PieChart;
import org.eazegraph.lib.models.PieModel;
import org.json.JSONException;
import org.json.JSONObject;

import kotlin.Unit;
import kotlin.jvm.functions.Function1;

public class MainActivity extends AppCompatActivity {
    MeowBottomNavigation bottomNavigation;
    TextView
    tvCases,tvRecovered,tvCritical,tvActive,tvTodayCases,tvTotalDeaths,tvTodayDeaths,tv
    AffectedCountries;
    SimpleArcLoader simpleArcLoader;
    ScrollView scrollView;
    PieChart pieChart;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        ConnectivityManager connectivityManager= (ConnectivityManager)
        getApplicationContext().getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo networkInfo=connectivityManager.getActiveNetworkInfo();
        if(networkInfo == null || !networkInfo.isConnected() || !networkInfo.isAvailable()){
            Dialog dialog=new Dialog(this);
            dialog.setContentView(R.layout.alert);
            dialog.setCancelable(false);

            dialog.getWindow().setLayout(WindowManager.LayoutParams.WRAP_CONTENT,Window
            Manager.LayoutParams.WRAP_CONTENT);
            dialog.getWindow().getAttributes().windowAnimations=
            android.R.style.Animation_Dialog;
            Button button =dialog.findViewById(R.id.btn);
            button.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    recreate();
                }
            });
        }
    }
}
```

```
        dialog.show();

    }
    else {
        tvCases = findViewById(R.id.tvCases);
        tvRecovered = findViewById(R.id.tvRecovered);
        tvCritical = findViewById(R.id.tvCritical);
        tvActive = findViewById(R.id.tvActive);
        tvTodayCases = findViewById(R.id.tvTodayCases);
        tvTotalDeaths = findViewById(R.id.tvTotalDeaths);
        tvTodayDeaths = findViewById(R.id.tvTodayDeaths);
        tvAffectedCountries = findViewById(R.id.tvAffectedCountries);

        simpleArcLoader = findViewById(R.id.loader);
        scrollView = findViewById(R.id.scrollStats);
        pieChart = findViewById(R.id.piechart);
        bottomNavigation=findViewById(R.id.bottom_navigation);
        bottomNavigation.add(new
MeowBottomNavigation.Model(1,R.drawable.ic_search));
        bottomNavigation.add(new
MeowBottomNavigation.Model(2,R.drawable.ic_home));
        bottomNavigation.add(new
MeowBottomNavigation.Model(3,R.drawable.ic_user));
        bottomNavigation.setOnClickMenuListener(new
Function1<MeowBottomNavigation.Model, Unit>() {
            @Override
            public Unit invoke(MeowBottomNavigation.Model model) {
                switch (model.getId()){
                    case 1:
                        startActivity(new
Intent(getApplicationContext(),AffectedCountries.class));
                        break;

                    case 2:
                        startActivity(new Intent(MainActivity.this,MainActivity.class));
                        finish();
                        break;

                    case 3:
                        startActivity(new Intent(getApplicationContext(),Creator.class));
                        break;
                }
            }
        });
    }
}
```

```
        }
        return null;
    }

    });

    fetchData();
}

private void fetchData() {

    String url = "https://corona.lmao.ninja/v2/all/";

    simpleArcLoader.start();

    StringRequest request = new StringRequest(Request.Method.GET, url,
        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {

                try {
                    JSONObject jsonObject = new JSONObject(response.toString());

                    tvCases.setText(jsonObject.getString("cases"));
                    tvRecovered.setText(jsonObject.getString("recovered"));
                    tvCritical.setText(jsonObject.getString("critical"));
                    tvActive.setText(jsonObject.getString("active"));
                    tvTodayCases.setText(jsonObject.getString("todayCases"));
                    tvTotalDeaths.setText(jsonObject.getString("deaths"));
                    tvTodayDeaths.setText(jsonObject.getString("todayDeaths"));
                    tvAffectedCountries.setText(jsonObject.getString("affectedCountries"));
                    pieChart.addPieSlice(new
PieModel("Cases", Integer.parseInt(tvCases.getText().toString()),
Color.parseColor("#FFA726")));
                    pieChart.addPieSlice(new
PieModel("Recoverd", Integer.parseInt(tvRecovered.getText().toString()),
Color.parseColor("#66BB6A")));
                    pieChart.addPieSlice(new
PieModel("Deaths", Integer.parseInt(tvTotalDeaths.getText().toString()),
```



```
Color.parseColor("#EF5350")));
        pieChart.addPieSlice(new
PieModel("Active", Integer.parseInt(tvActive.getText().toString()),
Color.parseColor("#29B6F6")));
        pieChart.startAnimation();

        simpleArcLoader.stop();
        simpleArcLoader.setVisibility(View.GONE);
        scrollView.setVisibility(View.VISIBLE);

    } catch (JSONException e) {
        e.printStackTrace();
        simpleArcLoader.stop();
        simpleArcLoader.setVisibility(View.GONE);
        scrollView.setVisibility(View.VISIBLE);
    }

    }, new Response.ErrorListener() {
@Override
public void onErrorResponse(VolleyError error) {
    simpleArcLoader.stop();
    simpleArcLoader.setVisibility(View.GONE);
    scrollView.setVisibility(View.VISIBLE);
    Toast.makeText(MainActivity.this, error.getMessage(),
Toast.LENGTH_SHORT).show();
}
});

RequestQueue requestQueue = Volley.newRequestQueue(this);
requestQueue.add(request);
}
public void goTrackCountries(View view) {
    startActivity(new Intent(getApplicationContext(), AffectedCountries.class));
}
}
```

## Chapter -5

### SNAPSHOTS



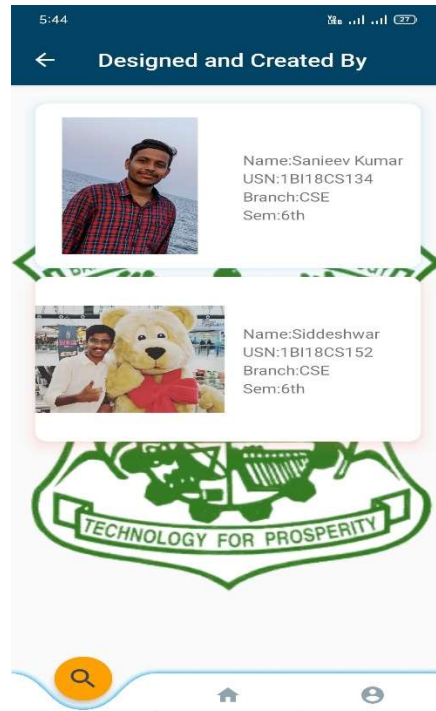
*Figure 5.1 Home Screen*



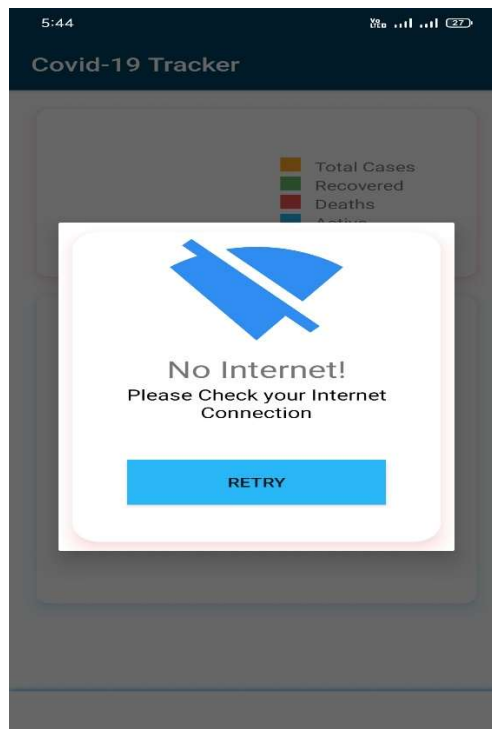
*Figure 5.2 Affected Countries*



*Figure 5.3 Details*



**Figure 5.4 Creator**



**Figure 5.5 Alert Dialog Box**

## Chapter -6

### CONCLUSION

“A Covid-19 Tracker” project was done in a perspective of understanding the Android development toolkit. Through this project, we have acquired a much deeper knowledge of the REST API and Design as a whole.

We thus would like to emphasize the importance of this project to many other perspectives of Technical, graphical and software concepts which we were unaware of.

#### 6.1 Future Enhancements

- In future the same project can be enhanced in such a way that we can interact more with the project. Also the project can be added more features.
- A vast amount of future work can be possible by following investigations and strategies.
- More features can be included and can be modified in a more versatile way.

## **BIBLIOGRAPHY**

### **Reference Books**

- [1] Erik Hellman, “Android Programming – Pushing the Limits”, 1st Edition, Wiley India Pvt Ltd, 2014. ISBN-13: 978-8126547197
- [2] Dawn Griffiths and David Griffiths, “Head First Android Development”, 1st Edition, O’Reilly SPD Publishers, 2015. ISBN-13: 978-9352131341
- [3] Bill Phillips, Chris Stewart and Kristin Marsicano, “Android Programming: The Big Nerd Ranch Guide”, 3 rd Edition, Big Nerd Ranch Guides, 2017. ISBN-13: 978-0134706054

### **Websites**

- [1] <https://developer.android.com/docs>
- [2] <https://corona.lmao.ninja/>