VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI-590018

A Mini Project report

on

"SCIENTIFIC CALCULATOR"

Submitted to the Visvesvaraya University, Belagavi
In partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering
In

COMPUTER SCIENCE AND ENGINEERING

Submitted By

MD MUSABBIR (1SP19CS053)
MONISHA.D (1SP19CS057)

Under the guidance

Of

Mr K A NAGABHIRAVANATH
Assistant Professor,
Dept. of CSE

Department of Computer Science and Engineering

S.E.A.COLLEGE OF ENGINEERING AND TECHNOLOGY

BENGALURU-560049

2021-2022

S.E.A COLLEGE OF ENGINEERING AND TECHNOLOGY

EktaNagar, Basavanpura, Virgonagar Post, K.R.Puram, Bengaluru, Karnataka 560059

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify the project work entitled "**ANDROID ATTENDENCE SYSTEM**" has been successfully carried out **Md Musabbir (1SP19CS053),Monisha D(1SP19CS057)** of 6^{TH} sem in partial fulfilment for the award of Bachelor of Engineering in Computer Science and Technology of the Visvesvaraya Technological University, Belgaum during the year 2020-21.The project report has been approved as it satisfied the academic requirement in respect of the mini project work prescribed for Bachelor of Engineering.

Signature of Guide

Prof. K A NAGABHAIRAVANATH

Signature of HOD

Dr. B.LOGANAYAGI

Signature of Internal Examiner

Signature of External Examiner

ACKNOWLEDGEMENT

Firstly, I thank the Management late **SHRIA KRISHNAPPA**, Chairman SEA College of Engineering and Technology for Providing Necessary infrastructure and creating good environment.I would like to express my profound thanks to our respected principal **Dr.B VENKETANARAYANA**for the encouragement and support given by him.I would like to express my profound thanks to our respected director. I would like to express my sincere thanks to our respected **Dr. B.LOGANAYAGI,HOD OF COMPUTER SCIENCE AND ENGINEERING** department,for her assistance and guidance.I'm thankful for the support rendered by my Project guide and coordinator **Prof. K A Nagabhairavanath** for his valuable suggestions.

I am also obliged, to the faculty members of CSE Department who rendered their valuable assistancefor the Project.And, I would like to express my heart full gratitude to my parents who have extended their helpthroughout my Project.And finally, I would like to express my heart full gratitude to my friends and all those who haveextended their help throughout my Project.

MD MUSABBIR (1SP19CS053)

MONISHA.D (1SP19CS057)

ABSTRACT

The process covers only of two approaches used in producing the simple calculator which are logic simulation and algorithm simulation due to the fact that it is difficult to develop a scientific simple calculator that will calculate every existing calculation in all field of study, this researchhas been limited to arithmetic operation calculation.Since it is time saying that there is no smoke without fire ,the effort to carry out the research was not first conceived. To particularilize the matter and concept 'what a calculator cando in solving tedious problems and how it is done? There is serious need for this study.In view of the aforementioned problems inherent in the design of the simple scientific calculator, the purpose of the study is to provide on overview of the simple design calculator.

INDEX

FIGURE INDEX

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, partnersand employeesdemand .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

A Scientific Calculator is useful for situations where need to calculate complex things like logs or trignometry.In such cases, the normal calculator wont be useful for us.Therefore , we are here to develop a scientific calculator.

The scientific calculator is a type of electronic calculator in which different complex calculating methods are involved.These methods involve mathemathical,scientific and some methods related to engineering. The functions involved in scientific calculator are scientific notations, floating point values, trignometric functions, logarithmic functions,fraction,factorial etc.

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 cashes.
- 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

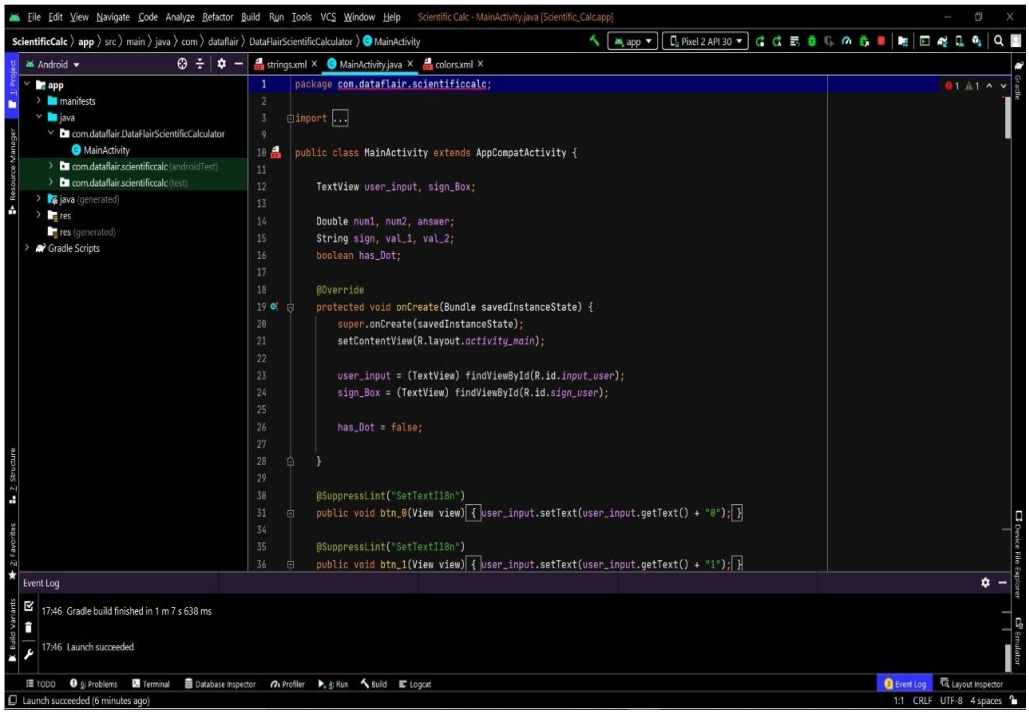


Figure 2.3: The Android Studio main window.

- 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

- This work was centered on the design and implementation of a simple scientific calculator for educational organization.
- The study traced calculator system as a tool to completely change mathematical knowledge and sophisticated problems solving strategies had advanced the field of simulated engine in mathematic.
- This project work also focused principally on numbers and arithmetic operations.
- The researcher investigated the manual system in detail with a view to finding out to automate the system.
- Interestingly, the end of simple calculator system was its ability to process number and operators, and provides a useful result.
- Therefore, this project will help immensely in following ways. Easy calculating of tedious mathematical problems,easy to retrival of errors and it will also be of a good assistance to any researcher on these topics.

3.2. Data Flow Diagram

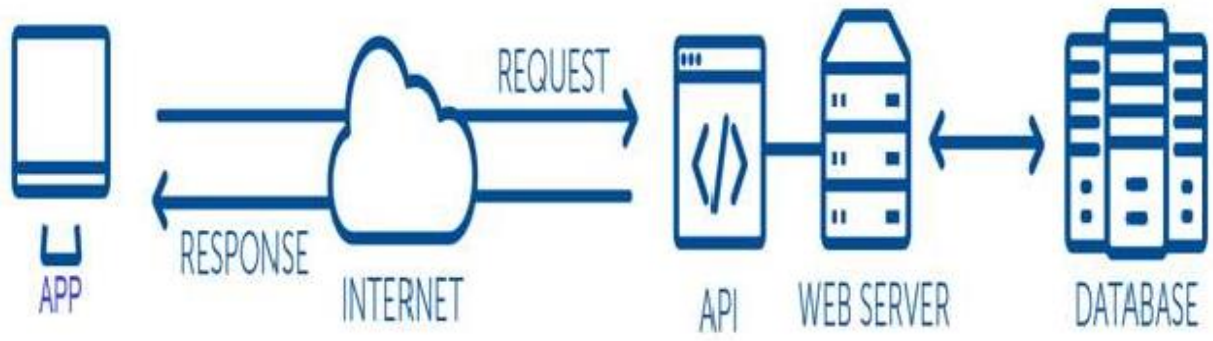


Figure 3.2: Data Flow Diagram

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 servic s.

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 real-time 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

- **Scroll View**

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 Linear Layout, 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

- **Card View**

Card View 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, Card View 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 Image View 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).

- **Text View**

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

- **Image view**

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

- **Edit text**

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.

- **Recycler View**

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 lighweight library which is easy to use and highly customizeable 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 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.

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 OperatingSystem 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()**

The 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.

- **privateSwipeRefreshLayout .**

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 width, we must specify its input type in inputType property of EditText which configures the keyboard according to input. EditText uses TextWatcherinterface to watch change made over EditText. For doing this, Edit Text 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 here 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

SOURCE CODE

6.1 ANDROID TEST

```
package com.halil.ozel.covid19stats;

import android.content.Context;

import androidx.test.InstrumentationRegistry;

import androidx.test.runner.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.getTargetContext();

        assertEquals("com.halil.ozel.covid19stats", appContext.getPackageName());

    }

}
```

6.2 MAIN (XML ACTIVITY DETAILS)

```
<?xml version="1.0" encoding="UTF-8"?>

<LinearLayout tools:context=".ui.activity.DetailActivity" android:orientation="vertical"
android:layout_height="match_parent" android:layout_width="match_parent"
```

xmlns:tools="http://schemas.android.com/tools"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:android="http://schemas.android.com/apk/res/android"><androidx.core.widget.NestedScr
ollView android:layout_height="match_parent"
android:layout_width="match_parent"><LinearLayout android:orientation="vertical"
android:layout_height="match_parent" android:layout_width="match_parent"
tools:layout_editor_absoluteY="32dp" tools:layout_editor_absoluteX="137dp"
android:layout_marginTop="10dp"><ImageView android:layout_height="wrap_content"
android:layout_width="match_parent" android:layout_gravity="center"
android:id="@+id/ivCountryPoster"/><TextView android:layout_height="wrap_content"
android:layout_width="wrap_content" android:layout_marginTop="5dp"
android:layout_gravity="center" android:id="@+id/tvCountryName"
android:textSize="20sp"/><LinearLayout android:orientation="horizontal"
android:layout_height="wrap_content" android:layout_width="wrap_content"
android:layout_marginTop="15dp" android:layout_marginStart="15dp"><ImageView
android:layout_height="wrap_content" android:layout_width="wrap_content"
android:id="@+id/ivTodayCases" android:background="@drawable/cases"/><TextView
android:layout_height="wrap_content" android:layout_width="wrap_content"
android:layout_gravity="center" android:textSize="20sp" android:layout_marginStart="10dp"
android:text="Today Cases : " android:layout_marginEnd="10dp"/><TextView
android:layout_height="wrap_content" android:layout_width="wrap_content"
android:layout_gravity="center" android:id="@+id/tvTodayCases" android:textSize="20sp"
android:layout_marginStart="10dp" android:layout_marginEnd="10dp"
android:textStyle="bold"/></LinearLayout><LinearLayout android:orientation="horizontal"
android:layout_height="wrap_content" android:layout_width="wrap_content"
android:layout_marginTop="15dp" android:layout_marginLeft="15dp"><ImageView
android:layout_height="wrap_content" android:layout_width="wrap_content"
android:id="@+id/ivTodayDeath" android:background="@drawable/death"/><TextView
android:layout_height="wrap_content" android:layout_width="wrap_content"
android:layout_gravity="center" android:textSize="20sp" android:layout_marginStart="10dp"
android:text="Today Death : " android:layout_marginEnd="10dp"/><TextView
android:layout_height="wrap_content" android:layout_width="wrap_content"
android:layout_gravity="center" android:id="@+id/tvTodayDeath" android:textSize="20sp"
android:layout_marginStart="10dp" android:layout_marginEnd="10dp"
android:textStyle="bold"/></LinearLayout><LinearLayout android:orientation="horizontal"
android:layout_height="wrap_content" android:layout_width="wrap_content"
android:layout_marginTop="15dp" android:layout_marginLeft="15dp"><ImageView
android:layout_height="wrap_content" android:layout_width="wrap_content"
android:id="@+id/ivTotalTest" android:background="@drawable/test"/><TextView
android:layout_height="wrap_content" android:layout_width="wrap_content"
android:layout_gravity="center" android:textSize="20sp" android:layout_marginStart="10dp"
android:text="Total Test : " android:layout_marginEnd="10dp"/><TextView
android:layout_height="wrap_content" android:layout_width="wrap_content"

android:layout_gravity="center" android:id="@+id/tvTotalTests" android:textSize="20sp"
android:layout_marginStart="10dp" android:layout_marginEnd="10dp"
android:textStyle="bold"/></LinearLayout><LinearLayout android:orientation="horizontal"
android:layout_height="wrap_content" android:layout_width="wrap_content"
android:layout_marginTop="15dp" android:layout_marginLeft="15dp"><ImageView
android:layout_height="wrap_content" android:layout_width="wrap_content"
android:id="@+id/ivTotalCases" android:background="@drawable/totalcases"/><TextView
android:layout_height="wrap_content" android:layout_width="wrap_content"
android:layout_gravity="center" android:textSize="20sp" android:layout_marginStart="10dp"
android:text="Total Cases : " android:layout_marginEnd="10dp"/><TextView
android:layout_height="wrap_content" android:layout_width="wrap_content"
android:layout_gravity="center" android:id="@+id/tvTotalCases" android:textSize="20sp"
android:layout_marginStart="10dp" android:layout_marginEnd="10dp"
android:textStyle="bold"/></LinearLayout><LinearLayout android:orientation="horizontal"
android:layout_height="wrap_content" android:layout_width="wrap_content"
android:layout_marginTop="15dp" android:layout_marginLeft="15dp"><ImageView
android:layout_height="wrap_content" android:layout_width="wrap_content"
android:id="@+id/ivTotalDeaths" android:background="@drawable/totaldeath"/><TextView
android:layout_height="wrap_content" android:layout_width="wrap_content"
android:layout_gravity="center" android:textSize="20sp" android:layout_marginStart="10dp"
android:text="Total Deaths : " android:layout_marginEnd="10dp"/><TextView
android:layout_height="wrap_content" android:layout_width="wrap_content"
android:layout_gravity="center" android:id="@+id/tvTotalDeaths" android:textSize="20sp"
android:layout_marginStart="10dp" android:layout_marginEnd="10dp"
android:textStyle="bold"/></LinearLayout><LinearLayout android:orientation="horizontal"
android:layout_height="wrap_content" android:layout_width="wrap_content"
android:layout_marginTop="15dp" android:layout_marginLeft="15dp"><ImageView
android:layout_height="wrap_content" android:layout_width="wrap_content"
android:id="@+id/ivTotalRecovered"
android:background="@drawable/liferecovered"/><TextView
android:layout_height="wrap_content" android:layout_width="wrap_content"
android:layout_gravity="center" android:textSize="20sp" android:layout_marginStart="10dp"
android:text="Total Recovered : " android:layout_marginEnd="10dp"/><TextView
android:layout_height="wrap_content" android:layout_width="wrap_content"
android:layout_gravity="center" android:id="@+id/tvTotalRecovered" android:textSize="20sp"
android:layout_marginStart="10dp" android:layout_marginEnd="10dp"
android:textStyle="bold"/></LinearLayout></LinearLayout></androidx.core.widget.NestedScro
llView></LinearLayout>

6.3 MAIN API (CORONA SERVICE)

```
package com.halil.ozel.covid19stats.api;

import com.halil.ozel.covid19stats.data.AllCountriesResponse;
```

```
import com.halil.ozel.covid19stats.data.CountriesResponse;
import java.util.List;
import retrofit2.Call;
import retrofit2.http.GET;
import retrofit2.http.Path;
public interface CoronaService {
    @GET("countries/?sort=country")
    Call<List<CountriesResponse>> getCountries();
    @GET("countries/{country}")
    Call<Countries Response>getCountryInfo(
            @Path("country") String country
    );
    @GET("all")
    Call<AllCountriesResponse> getAllCountries();
}
```

6.4 CORONA INFOMATION

```
package com.halil.ozel.covid19stats.data;
public class CountryInfo {
    private String flag;
    private String _id;
    private String iso2;
    private String iso3;
    public String getFlag() {
        return flag;
    }
    public void setFlag(String flag) {
        this.flag = flag;
    }
```

```
    public String get_id() {
        return _id;
    }
    public void set_id(String _id) {
        this._id = _id;
    }
    public String getIso2() {
        return iso2;
    }
    public void setIso2(String iso2) {
        this.iso2 = iso2;
    }
    public String getIso3() {
        return iso3;
    }
    public void setIso3(String iso3) {
        this.iso3 = iso3;
    }
}
```

6.5 TEST

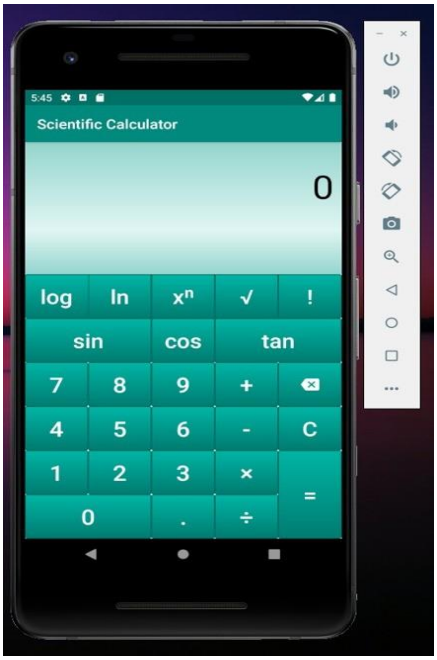
```
package com.halil.ozel.covid19stats;
import org.junit.Test;
import static org.junit.Assert.*;
/**
* Example local unit test, which will execute on the development machine (host).
*
* @see <a href="http://d.android.com/tools/testing">Testing documentation</a>
*/
```

```
public class ExampleUnitTest {
    @Test
    public void addition_isCorrect() {
        assertEquals(4, 2 + 2);
    }
}
```

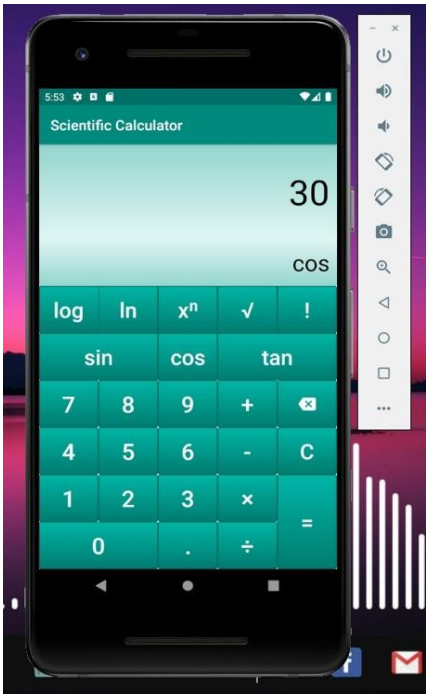
CHAPTER 7

SNAP SHOTS

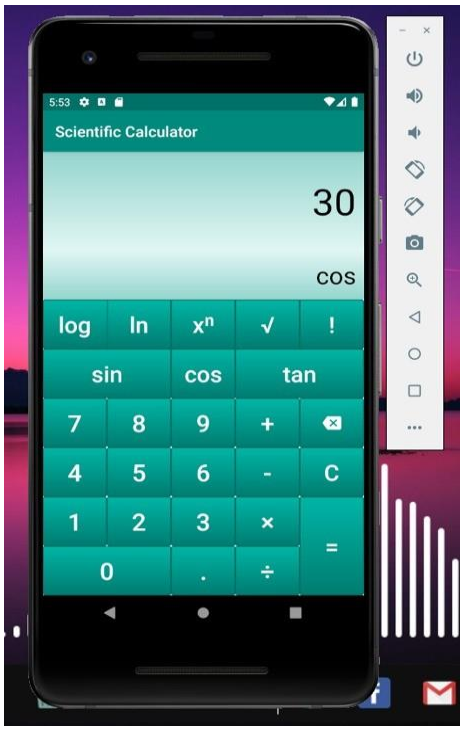
7.1 HOME PAGE



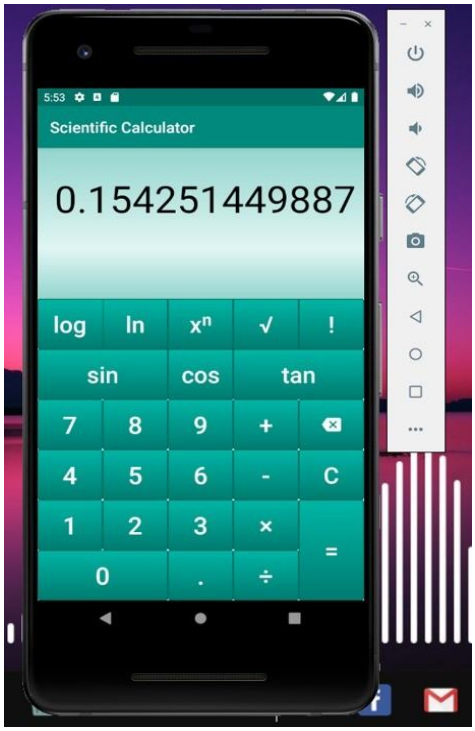
7.2 CALCULATION 1



7.3 CALCULATION 2



7.4 CALCULATION 3



CONCLUSION

A scientific calculator is a type of electronic calculator, usually but not always handheld, designedswith the financial calculator market.

REFERENCES

WEBSITES

- .in.searchley.com/MobileDevelop

- www.google.com

TEXTBOOKS

- Android Developer Fundamentals Course – By Google.
- Android Programming – Pushing The Limits – By Erik Hellman.
- Head First Android Development – By Dawn Griffiths and David Griffiths.