CHAPTER 1

INTRODUCTION

Android dominates the mobile OS industry because of the long list of features it provides. It's user-friendly, has huge community support, provides a greater extent of customization, and a large number of companies build Android compatible smartphones.

As a result, the market observes a sharp increase in the demand for developing Android mobile applications, and with that companies need smart developers With the right skill set. At first, the purpose of Android was thought of as a mobile operating system. However, with the advancement of code libraries and its popularityamong developers of the divergent domain, Android becomes an absolute set of software for all devices like tablets, wearables, set-top boxes, smart TVs, notebooks, etc.

Several mobile fitness devices as well as smart watches have emerged on the technology landscape. However, the rate of adoption of these devices is still low especially in developing countries with a teeming population. On the other hand, smart phones are becoming ubiquitous given their steady price decline.

To this end, the present study aims to leverage the smartphone platform by Developing a smart phone fitness app that tracks the calories of individuals who go about their daily activities while carrying their smart phones with them.

1.1 PROBLEM STATEMENT

A problem statement for a calorie calculator Android development project could be something like this: “Many people struggle to maintain or lose weight and need a way to easily track their daily caloric intake. A Calorie Calculator & Suggester Android System Application can ensure the user to have a proper intake of calories required for their daily diet. This application allows users to have a good and balanced diet1.”

1.2 OBJECTIVE

- Mention the user height
- Mention the user weight
- Mention the user age
- Calculate the calorie accordingly
- Count the number of calories consumed per day

CHAPTER 2

LITERATURE SURVEY

2.1 TECHNOLOGICAL DETAILS

➢ ANDROID STUDIO: IDE

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems or as a subscription-based service in 2020. It is a replacement for the Eclipse Android Development Tools (E-ADT) as the primary IDE for native Android application development.

Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0.

On May 7, 2019, Kotlin replaced Java as Google's preferred language for Android app development. Java is still supported, as is C++.

Feature:
A specific feature of the Android Studio is an absence of the possibility to switch auto save feature off.

The following features are provided in the current stable version:

- Gradle-based build support
- Android-specific refactoring and quick fixes
- Lint tools to catch performance, usability, version compatibility and other problems
- Pro Guard integration and app-signing capabilities

- Template-based wizards to create common Android designs and components
- A rich layout editor that allows users to drag-and-drop UI components, option top reviewlayouts on multiple screen configurations Support for building Android Wear apps
- Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine
- Android Virtual Device (Emulator) to run and debug apps in the Android studio.

➢ JAVA PROGRAMMING

Android applications are developed using the Java language. As of now, that's really your only option for native applications. Java is a very popular programming language developed by Sun Microsystems (now owned by Oracle). Developed long after C and C++,Java incorporates many of the powerful features of those powerful languages while addressing some of their drawbacks. Still, programming languages are only as powerful as their libraries. These librariesexist to help developers build applications.
Some ofthe Java's important core features are:

- It's easyto learn and understand
- It's designed to be platform-independent and secure, using
- virtual machines
- It's object-oriented

Android relies heavily on these Java fundamentals. The Android SDK includes many standard Java libraries (data structure libraries, math libraries, graphics libraries, networking libraries and everything else you could want) as well as special Android libraries that will help you develop awesome Android applications.

➢ XML BASICS

Android layouts are written in eXtensible Markup Language, also known as XML. Much like HTML (or HyperText Markup Language), XML is also a markup language. It was createdas a standard way to encode data in internet-based applications. However, unlike HTML, XMLis case-sensitive, requires each tag is closed properly, and preserves whitespace.

Much like creating an HTML layout and later altering it with jQuery, as we've done in previous courses, we can create XML layouts in Android, and later alter them using Java logic.

Android XML layouts are also part of a larger umbrella of Android files and components called resources. Resources are the additional files and static content an application needs, such as animations, color schemes, layouts, menu layouts.

➢ OBJECT ORIENTED PROGRAMMING LANGUAGE

OOP is a programming style or technique that relies upon the definition of data structures called objects. For those new to OOP, an object can be thought of much like a custom data type. For example, you might have a Dog object, which represents the blueprint fora generic dog, with a name, breed, and gender. You could then create different instances of the Dog object to represent specific dogs. Each Dog object must be created by calling its constructor (a method that has thesame name as the object itself, and may or may not have parameters for setting initial values). For example, the following Dog objects use a constructor with three parameters(name, breed, gender):

```
Dog dog1 = new Dog("Lassie", collie, female);
Dog dog2 = new Dog("Fifi", poodle, female);
Dog dog3 = new Dog("Asta", foxterrier, male);
```

So where is this Dog object defined? Well, here we need to begin defining some of the fundamental building blocks of the Java programming language. A class provides a definition for an object. Therefore, there is a Dog class somewhere—either defined by you or in some library somewhere. Generally speaking, a class will be defined in its own file, with the filename matching the class name (e.g. Dog.java). There are exceptions to this rule, such as classes defined within other classes (when a class is declared within a class, it is generally defined for use within the parent class only as a helper class, and referred to as an inner class).

When you want to reference an object from within another class, you need to include an import statement in the top of your class file, much like you would use a #include statement in a compiled language like C. A class typically describes the data and behavior of an object. The behavior is defined using class methods. Method is the common term for a subroutine in an OOP language. Many common object classes are defined in shared class libraries like software development kits (SDKs), whereas others are defined by you, the developer, for your own purposes. Software is then built up by using and manipulating object instances in different ways

CHAPTER 3

SYSTEM REQUIREMENT AND SPECIFICATION

3.1 HARDWARE REQUIREMENTS

- 64-bit Microsoft® Windows® 8/10
- x86_64 CPU architecture; 2nd generation Intel Core or newer, or AMDCPU with support for a Windows Hypervisor
- 8 GB RAM or more
- 8 GB of available disk space minimum (IDE + Android SDK + Android Emulator)
- 1280 x 800 minimum screen resolution, Android studio version 4.2.2

3.2 SOFTWARE REQUIREMENTS

• The AOSP master branch is traditionally developed and tested on Ubuntu Long Term Support (LTS) releases.

But other distributions may be used. See Establishing a Build Environment form additional required packages and the commands to install them.

• Your workstation must have the software listed below.

• These requirements apply to the AOSP master branch. For Android versions 8.0 (Oreoor O) through 5.0 (Lollipop or L)

• Consider using the included Dockerfile to ease installation of all required packages .

• For the manual method, see Supporting Older Versions.

CHAPTER 4

SYSTEM DESIGN

Any App Starting needs AndroidManifest.XML file to start. And any new project content will automatically generate an AndroidManifest.XML file. The systems will automatically looking for a logo in Android Manifest to react the corresponding operation when any component of the program triggers events. To define the system, the first thing is launching the Activity: Android Activity. Setting the action and category realize the switch between different Activities. When any components of the program is about to use, declaration must be in the Android Manifest. Xml files. To be clear that authorities must be illustrated as the statement of provider. Each component has a lot of attributes; the program will define different attributes according to different needs.

The basic structure content of Android project includes: the SRC (source code), gen (constant that Android system automatically generates), res (resource file), and the layout of file and pictures in the main storage program interface. Every Android interface is a visual interface, which has its unique layout configuration files. We can configure various layout and resources files according to the requirements, such as images, text and color reference, which can form different visual interface and glaring effect.

4.1 DATA FLOW DIAGRAM

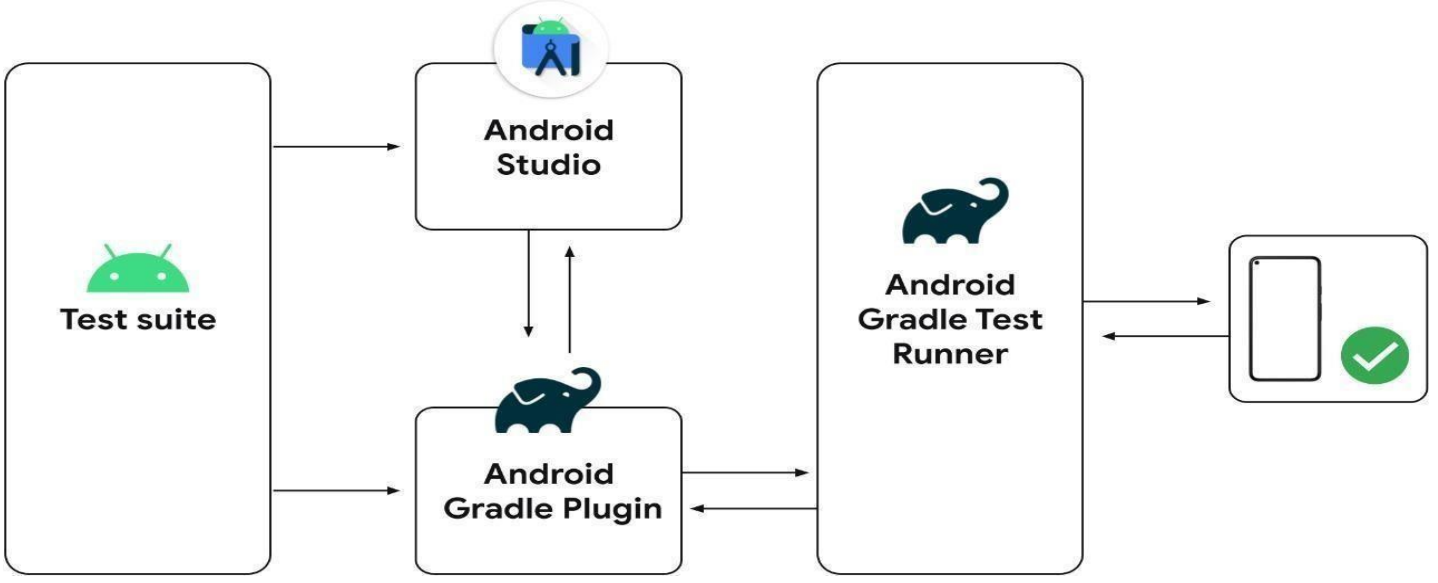


Fig 4.1: Data Flow Diagram of Android studio

A data flow diagram (DFD) maps out the flow of information for any process or system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both. It shows how data enters and leaves the system, what changes the information, and where data is stored .

The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart. It uses defined symbols like rectangles, circles and arrows, plus short textlabels, to show data inputs, outputs, storage points and the routes between each destination. Dataflowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-levelDFDs that dig progressively deeper into how the data is handled.

CHAPTER 5

IMPLEMENTATION

Calories Calculator Android App Project is an native android application meant to leverage the smartphone platform by developing a smart phone fitness app that tracks the calories of individuals who go about their daily activities while carrying their smart phones with them.

Mainactivity.java

```
package com.example.caloriecalculatorproject;

import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.AppCompatButton;
import android.os.Bundle;
import android.view.View;
import android.widget.RadioGroup;
import android.widget.TextView;
import com.google.android.material.radiobutton.MaterialRadioButton;
import com.google.android.material.textfield.TextInputEditText;
import java.util.regex.Pattern;
// Creating the main class and extending it with the AppCompatActivity class
public class MainActivity extends AppCompatActivity {
// Declaring the required variables
private TextInputEditText age, height, weight;
private RadioGroup gender;
private MaterialRadioButton male, female;
private TextView calories, required, textView1, textView2, textView3, textView4, textView5, textView6,
text_dummy;
private AppCompatButton calculate, reset;
// Creating the onCreate method
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 // Setting the content view to the activity_main.xml file
 setContentView(R.layout.activity_main);
 // Initializing the variables
 age =  findViewById(R.id.age);
 height = findViewById(R.id.height);
 weight = findViewById(R.id.weight);
 gender = findViewById(R.id.gender);
 male = findViewById(R.id.male);
 female = findViewById(R.id.female);
 calories = findViewById(R.id.calories);
 textView1 = findViewById(R.id.textView1);
 textView2 = findViewById(R.id.textView2);
 textView3 = findViewById(R.id.textView3);
 textView4 = findViewById(R.id.textView4);
 textView5 = findViewById(R.id.textView5);
 textView6 = findViewById(R.id.textView6);
 text_dummy = findViewById(R.id.text_dummy);
 required = findViewById(R.id.required);
 calculate = findViewById(R.id.calculate);
 reset = findViewById(R.id.reset);
 // Creating the onClickListener for the reset button
 // This will reset all the values to the default values
 reset.setOnClickListener(v -> {
    age.setText("");
    height.setText("");
    weight.setText("");
    gender.clearCheck();
    calories.setText("Calories");
    textView1.setText("");
    textView2.setText("");
    textView3.setText("");
```

```
        textView4.setText("");
        textView5.setText("");
        textView6.setText("");
        text_dummy.setVisibility(View.GONE);
        required.setVisibility(View.GONE);
    });

// Creating the onClickListener for the calculate button
    calculate.setOnClickListener(v -> {
        // Getting the values from the text fields
        String ageText = age.getText().toString();
        String heightText = height.getText().toString();
        String weightText = weight.getText().toString();
        // Creating the pattern for the regular expression
        // This will check if the value is a number or not
        Pattern pattern = Pattern.compile("[0-9]+");
        // Creating the variables for the checks and setting them to false
        // These will be used to check if the values are empty or not
        boolean ageCheck = false;
        boolean heightCheck = false;
        boolean weightCheck = false;
        // Checking if the age text field is empty or not
        // If it is empty, then it will show an error message
        if(ageText.isEmpty()){
            age.setError("Please enter your age");
            age.requestFocus();
            ageCheck = false;
        } else if (!pattern.matcher(ageText).matches()) {
            age.setError("Please enter your age correctly");
            age.requestFocus();
            ageCheck = false;
        } else {
            age.setError(null);
            ageCheck = true;
```

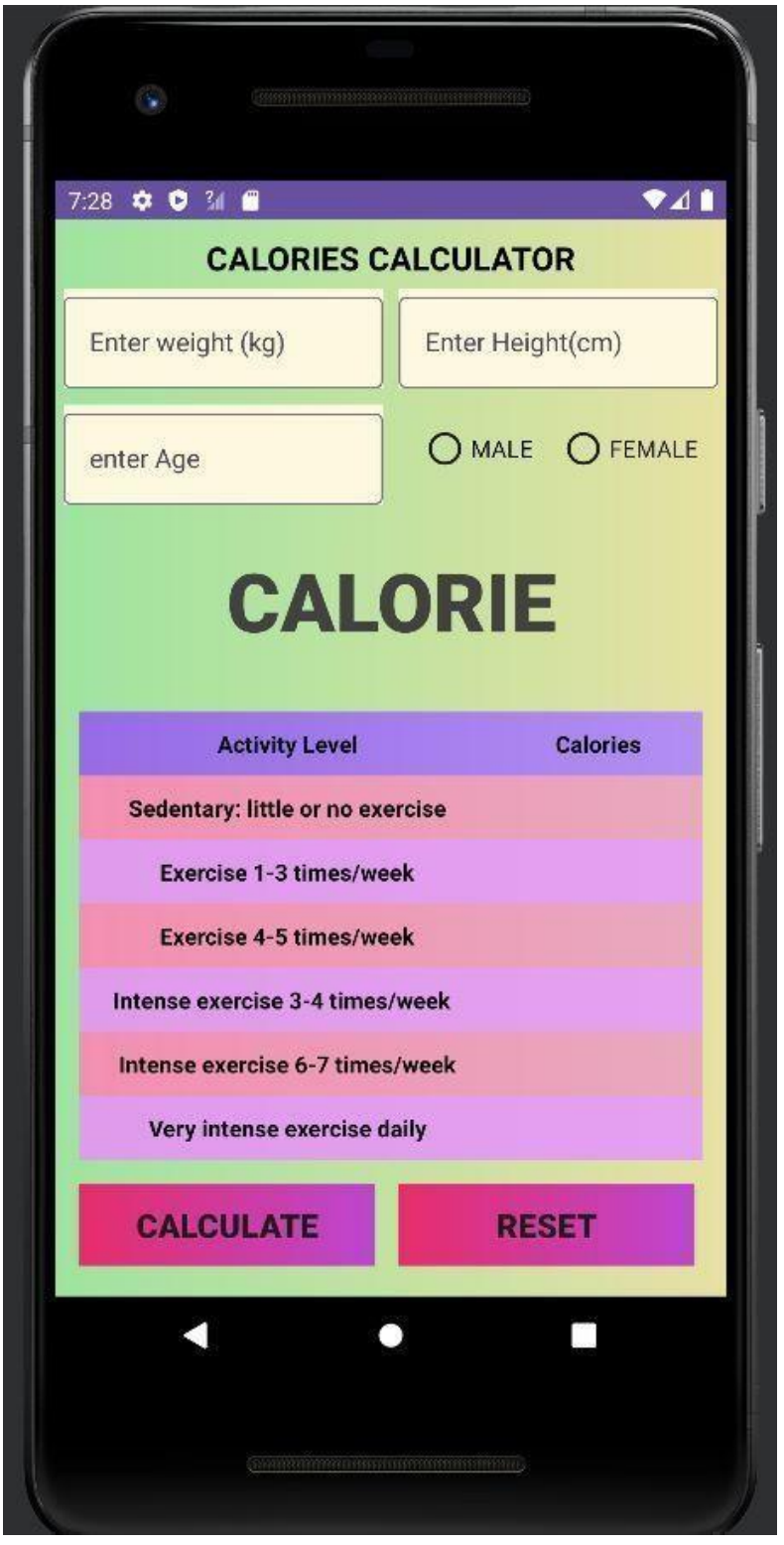
```
}
// Checking if the height text field is empty or not
// If it is empty, then it will show an error message
if(heightText.isEmpty()){
    height.setError("Please enter your height");
    height.requestFocus();
    heightCheck = false;
} else if (!pattern.matcher(ageText).matches()) {
    age.setError("Please enter your age correctly");
    age.requestFocus();
    heightCheck = false;
} else {
    height.setError(null);
    heightCheck = true;
}
// Checking if the weight text field is empty or not
// If it is empty, then it will show an error message
if(weightText.isEmpty()){
    weight.setError("Please enter your weight");
    weight.requestFocus();
    weightCheck = false;
} else if (!pattern.matcher(ageText).matches()) {
    age.setError("Please enter your age correctly");
    age.requestFocus();
    weightCheck = false;
} else {
    weight.setError(null);
    weightCheck = true;
}
// Checking if the user has selected the gender or not
if(gender.getCheckedRadioButtonId() == -1) {
    required.setText("Please Select Gender");
    required.setVisibility(View.VISIBLE);
} else {
```

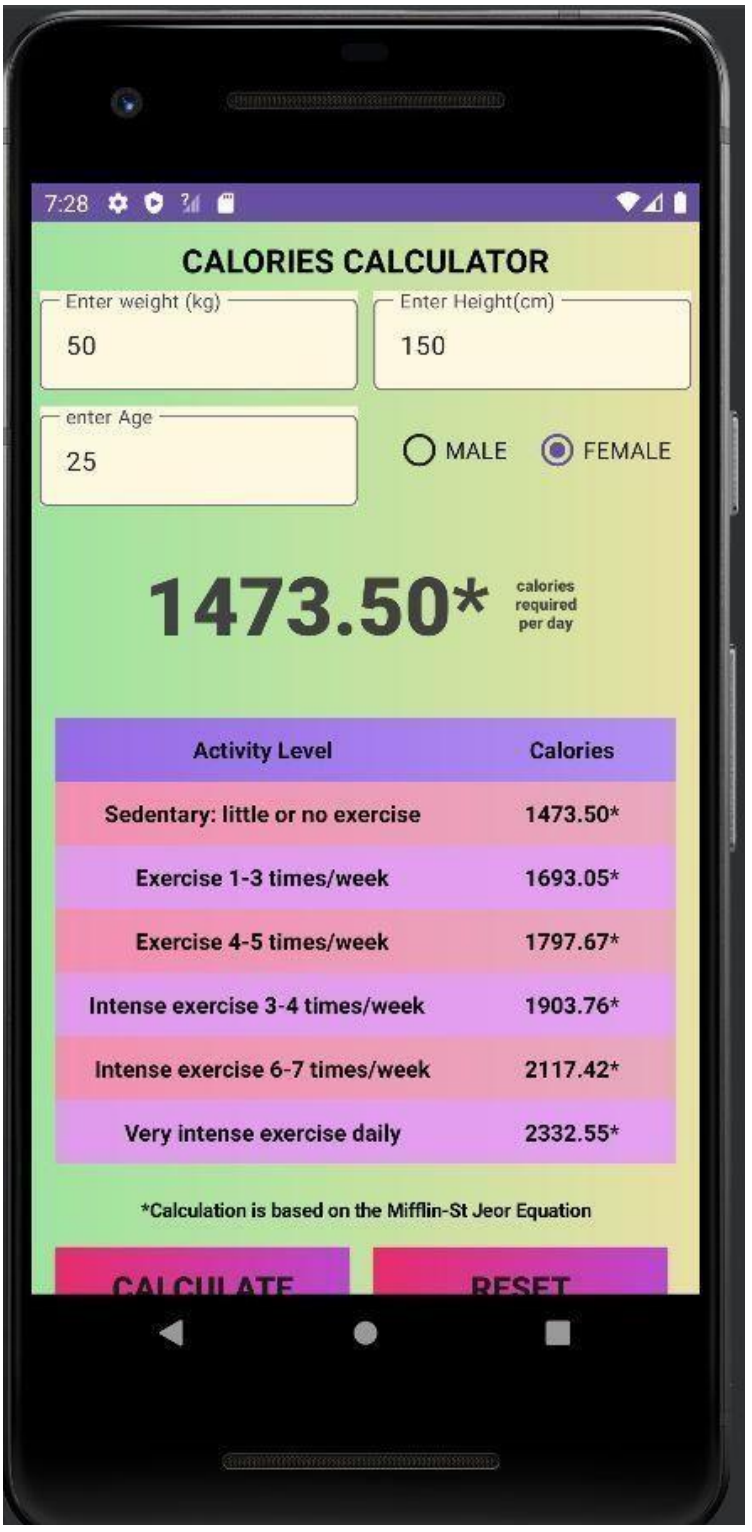
```
            required.setText("");
            required.setVisibility(View.GONE);

            // Checking if all the values are not empty
            if(ageCheck && heightCheck && weightCheck){

                // Calling the calculateBMR method
                calculateCalorie();
            }
        }
    });
}

// Creating the calculate method to calculate the calories required
public void calculateCalorie(){

    // Getting the values from the text fields
    int ageValue = Integer.parseInt(age.getText().toString());
    int heightValue = Integer.parseInt(height.getText().toString());
    int weightValue = Integer.parseInt(weight.getText().toString());

    // Creating the variable for the total calories
    double totalCalories = 0;

    if(gender.getCheckedRadioButtonId()== male.getId()){
        // If user is "Male" then the following formula will be used to calculate the calories
        totalCalories = (10 * weightValue) + (6.25 * heightValue) - (5 * ageValue + 5);

        // Setting the text to the calories text view
        text_dummy.setVisibility(View.VISIBLE);
    } else {
        // If user is "Female" then the following formula will be used to calculate the calories
        totalCalories = (10 * weightValue) + (6.25 * heightValue) - (5 * ageValue - 161);
        calories.setText(String.format("%.2f", totalCalories)+"*");
```

```
            text_dummy.setVisibility(View.VISIBLE);
        }

        // Setting the text to the calories in the table layout and rounding it to 2 decimal places
        textView1.setText(String.format("%.2f", totalCalories)+"*");
        textView2.setText(String.format("%.2f", totalCalories*1.149)+"*");
        textView3.setText(String.format("%.2f", totalCalories*1.220)+"*");
        textView4.setText(String.format("%.2f", totalCalories*1.292)+"*");
        textView5.setText(String.format("%.2f", totalCalories*1.437)+"*");
        textView6.setText(String.format("%.2f", totalCalories*1.583)+"*");

        // Setting the text to the text view and making it visible
        required.setText("*"+"Calculation is based on the Mifflin-St Jeor Equation");
        required.setTextSize(12);
        required.setVisibility(View.VISIBLE);
    }
}
```

CHAPTER 6

RESULT




7:28
CALORIES CALCULATOR
Enter weight (kg)
50
Enter Height(cm)
150
enter Age
25
MALE
FEMALE
1473.50*
calories required per day
Activity Level
Calories
Sedentary: little or no exercise
1473.50*
Exercise 1-3 times/week
1693.05*
Exercise 4-5 times/week
1797.67*
Intense exercise 3-4 times/week
1903.76*
Intense exercise 6-7 times/week
2117.42*
Very intense exercise daily
2332.55*
*Calculation is based on the Mifflin-St Jeor Equation
CALCULATE
RESET

CHAPTER 7

CONCLUSION

We believe this app will help many people who want to take care of their fitness but cannot afford fitness devices such as smart watches or any other fitness devices. This app can be downloaded in any smartphone with basic internet access. The app is user friendly and is easyfor anyone to understand the working. The UIis attractive to the users and it is a simple processas well. It works in most of the android phones.

Using this app, people from all levels will have access to fitness monitoring and they can keeps check on their health as well. This will improve the mortality rate of humans and most importantly, the target audience is larger, broader, and diverse.

We hope that one day fitness will not be restricted to just the wealthy class of people but to all people who want to maintain their health.

7.1 FUTURE ENHANCEMENT

A number of mobile fitness devices as well as smart watches have emerged on the technology landscape. However, the rate of adoption of these devices is still low especially in developing countries with a teeming population. On the other hand, smart phones are becoming ubiquitous given their steady price decline. To this end, the present study aims to leverage the smartphone platform by developing a smart phone fitness app that tracks the calories burnt by individuals who go about their daily activities while carrying their smart phones with them..

BIBLIOGRAPHY

1) http://www.google.com

2) http://www.w3school.com

3) http://www.tutorialspoint.com

4) http://www.wikipidea.com

CHAPTER 8

APPENDIX

SOURCE CODE

8.1 Manifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
  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="@drawable/gradiant"
  android:orientation="vertical"
  tools:context=".MainActivity">

<LinearLayout
 android:layout_width="match_parent"
 android:layout_height="match_parent"
 android:orientation="vertical">
 <com.google.android.material.textview.MaterialTex
    tViewandroid:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:layout_marginBottom="10dp"
    android:text="Calories Calculator"
    android:textAlignment="center"
    android:textAllCaps="true"
    android:textColor="@color/black"
    android:textSize="20sp"
    android:textStyle="bold" />
 <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <com.google.android.material.textfield.TextInput
       Layoutandroid:layout_width="0dp"
       android:layout_height="wrap_content"
       android:layout_margin="5dp"
       android:layout_weight="1"
       android:hint="Enter weight (kg)">
       <com.google.android.material.textfield.TextInputEditText
          android:id="@+id/weight"
```

```
            android:layout_height="wrap_c
            ontent"
            android:inputType="numberDe
            cimal" />
        </com.google.android.material.textfield.TextInputLayout>
        <com.google.android.material.textfield.TextInp
            utLayoutandroid:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_margin="5dp"
            android:layout_weight="1"
            android:hint="Enter Height (cm)">
            <com.google.android.material.textfield.TextInputEditTe
                xtandroid:id="@+id/height"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:inputType="numberDecimal" />
        </com.google.android.material.textfield.TextInputLayout>
    </LinearLayout>
    <androidx.appcompat.widget.LinearLayoutCompat
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <com.google.android.material.textfield.TextInp
            utLayoutandroid:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_margin="5dp"
            android:layout_weight="1"
            android:hint="Enter Age">
            <com.google.android.material.textfield.TextInputE
                ditText android:id="@+id/age"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:inputType="numberDecimal" />
        </com.google.android.material.textfield.TextInputLayout>
        <RadioGroup
            android:id="@+id/ge
            nder"
            android:layout_width
            ="0dp"
            android:layout_height="wrap_content"
            android:layout_marginLeft="5dp"
            android:layout_marginTop="5dp"
            android:layout_marginRight="5dp"
            android:layout_weight="1"
            android:gravity="center"
            android:orientation="horizontal"
            android:paddingVertical="3dp">

            <com.google.android.material.radiobutton.MaterialRadio
                Button android:id="@+id/male"
```

```
            android:textColor="@color/black" />
        <com.google.android.material.radiobutton.MaterialRadio
            Button android:id="@+id/female"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="FEMALE"
            android:textAllCaps="true" />

    </RadioGroup>

</androidx.appcompat.widget.LinearLayoutCompat>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:gravity="center"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/cal
        ories"
        android:layout_width="wrap_
        content"
        android:layout_height="wrap
        _content"
        android:layout_marginStart="
        10dp"
        android:layout_marginTop="
        20dp"
        android:layout_marginEnd="
        10dp"
        android:layout_marginBotto
        m="20dp"
        android:fontFamily="sans-
        serif-black"
        android:gravity="center"
        android:text="Calorie"
        android:textAlignment="cent
        er"
        android:textAllCaps="true"
        android:textColor="@color/cardview_dark_background"
        android:textSize="50sp"
        android:textStyle="bold" />
    <TextView
        android:id="@+id/text_dumm
        y"
        android:layout_width="50dp"
        android:layout_height="70dp"
        android:layout_marginTop="2
        0dp"
        android:layout_marginEnd="
```

```
android:layout_width="match_par
ent"
android:layout_height="wrap_cont
ent"
android:layout_margin="5dp">
<TableLayout
  android:id="@+id/calorieTable"
  android:layout_width="match_paren
  t"
  android:layout_height="wrap_conten
  t"android:layout_margin="10dp"
  android:background="@android:color/white">
  <TableRow android:background="@drawable/gradiant2">
    <TextView
      android:layout_width="0dp"
      android:layout_height="wrap_content"
      android:layout_weight="4"
      android:gravity="center"
      android:padding="10sp"
      android:text="Calories"
      android:textColor="@color/white"
      android:textSize="18sp"
      android:textStyle="bold" />
    <TextView
      android:layout_width="0dp"
      android:layout_height="wrap_content"
      android:layout_weight="2"
      android:gravity="center"
      android:padding="10sp"
      android:text="Calories"
      android:textColor="@color/white"
      android:textSize="18sp"
      android:textStyle="bold" />
  </TableRow>
  <TableRow android:background="@drawable/gradiant3">
    <TextView
      android:layout_width="0dp"
      android:layout_height="wrap_content"
      android:layout_weight="4"
      android:gravity="center"
      android:padding="10sp"
      android:text="Sedentary: little or no
      exercise"android:textColor="@color/black"
      android:textSize="14sp"
      android:textStyle="bold" />
    <TextView
      android:id="@+id/textView1"
      android:layout_width="0dp"
      android:layout_height="wrap_cont
      ent"android:layout_weight="2"
```

```
            android:textSize="
            14sp"
            android:textStyle=
            "bold" />
        </TableRow>
        <TableRow android:background="@drawable/gradiant4">
            <TextView
            android:layout_width="0d
            p"
            android:layout_height="wrap_
            content"
            android:layout_weight="4"
            android:gravity="center"
            android:padding="10sp"
            android:text="Exercise 1-3
            times/week"
            android:textColor="@color/bla
            ck" android:textSize="14sp"
            android:textStyle="bold" />
            <TextView
            android:id="@+id/textView2
            "
            android:layout_width="0dp"
            android:layout_height="wrap
            _content"
            android:layout_weight="2"
            android:gravity="center"
            android:padding="10sp"
            android:textColor="@color/bl
            ack" android:textSize="14sp"
            android:textStyle="bold" />
        </TableRow>
        <TableRow android:background="@drawable/gradiant3">
            <TextView
            android:layout_width="0d
            p"
            android:layout_height="wrap_
            content"
            android:layout_weight="4"
            android:gravity="center"
            android:padding="10sp"
            android:text="Exercise 4-5
            times/week"
            android:textColor="@color/bla
            ck" android:textSize="14sp"
            android:textStyle="bold" />
            <TextView
            android:id="@+id/textView3
            "
            android:layout_width="0dp"
```

```
            android:layout_height="wrap_conte
            nt"android:layout_weight="4"
            android:gravity="center"
            android:padding="10sp"
            android:text="Intense exercise 3-4
            times/week "
            android:textColor="@color/black"
            android:textSize="14sp"
            android:textStyle="bold" />
        <TextView
            android:id="@+id/textView4"
            android:layout_width="0dp"
            android:layout_height="wrap_c
            ontent"
            android:layout_weight="2"
            android:gravity="center"
            android:padding="10sp"
            android:textColor="@color/blac
            k" android:textSize="14sp"
            android:textStyle="bold" />
    </TableRow>
    <TableRow android:background="@drawable/gradiant3">
        <TextView
            android:layout_width="0dp"
            android:layout_height="wrap_conte
            nt"android:layout_weight="4"
            android:gravity="center"
            android:padding="10sp"
            android:text="Intense exercise 6-7
            times/week"
            android:textColor="@color/black"
            android:textSize="14sp"
            android:textStyle="bold" />
        <TextView
            android:id="@+id/textView5"
            android:layout_width="0dp"
            android:layout_height="wrap_c
            ontent"
            android:layout_weight="2"
            android:gravity="center"
            android:padding="10sp"
            android:textColor="@color/blac
            k" android:textSize="14sp"
            android:textStyle="bold" />
    </TableRow>
    <TableRow android:background="@drawable/gradiant4">
        <TextView
            android:layout_width="0dp"
            android:layout_height="wrap_conte
            nt" android:layout_weight="4"
```

```
            android:textSize="14
            sp"
            android:textStyle="b
            old" />
          <TextView
            android:id="@+id/textView6"
            android:layout_width="0dp"
            android:layout_height="wrap_c
            ontent"
            android:layout_weight="2"
            android:gravity="center"
            android:padding="10sp"
            android:textColor="@color/blac
            k" android:textSize="14sp"
            android:textStyle="bold" />
  </TableRow>
</TableLayout>

    <TextView
      android:id="@+id/requir
      ed"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_below="@+id/calorieTab
      le"android:layout_marginTop="10dp"
      android:layout_marginBottom="10dp"
      android:gravity="center"
      android:text="All Fields are
      Required"
      android:textColor="@color/blac
      k" android:textSize="16sp"
      android:textStyle="bold"
      android:visibility="gone" />
  </RelativeLayout>
  <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_marginHorizontal="5dp"
    android:layout_marginBottom="10dp"
    android:layout_weight="4"
    android:gravity="center"
    android:orientation="horizontal">
    <androidx.appcompat.widget.AppCompatB
      uttonandroid:id="@+id/calculate"
      android:layout_width="110dp"
      android:layout_height="50dp"
      android:layout_marginLeft="10dp"
      android:layout_marginRight="15dp"
      android:layout_weight="2"
```

```
            app:cornerRadius="10dp" />
        <androidx.appcompat.widget.AppCo
            mpatButton
            android:id="@+id/reset"
            android:layout_width="110dp"
            android:layout_height="50dp"
            android:layout_marginRight="15dp
            " android:layout_weight="2"
            android:background="@drawable/g
            radiant5" android:onClick="reset"
            android:text="RESET"
            android:textSize="20sp"
            android:textStyle="bold" />
      </LinearLayout>
    </LinearLayout>
  </ScrollView>
```