StopWatch

A Minor Project Report

Submitted by

TUSHAR PANDEY

in partial fulfilment for the award of the degree

of

B.TECH

IN

COMPUTER SCEINCE At

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GOVIND BALLABH PANT GOVT. ENGINEERING COLLEGE, OKHLA

NEW DELHI

DECEMBER,2018

Acknowledgments

I take this opportunity to express my profound gratitude and deep regards to my guide Mr. Yogendra Kumar(Android Instructor, Kyrion Technologies), for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by him time to time shall carry me a long way in the journey of life on which I am about to embark.

I also take this opportunity to express a deep sense of gratitude to Mr Kapila Jain(Manager, Kyrion Technologies) for their cordial support, valuable information and guidance which helped me in completing this task through various stages from their organisation.

I am obliged to staff members of Hewlett Packard Enterprise Teaching and mentoring staff for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

Lastly, i would like to express my gratefulness to my parents and friends for having shown their trust and faith in me and boosting my morals constantly.

Abstract

The era of mobile technology opens the windows to the android app.Various instruments , mechanical tools and machines are vanishing and mobile phones are taking the lead and their place in this world with their increasing computing capability and new more capable sensors and circuits.Its time when people are leaving the habit of carrying different devices for their daily routine as these objects are continually being incorporated or virtually realised into a wall mobile phone which can fit into our pockets very easily.

A similar step in this direction is my Android™ application that virtually realises a Stopwatch. Earlier and even until very recent times, StopWatches were a device, a bulky piece of metal machinery that was essentially an analog clock, handheld and was used to measure time durations with slightly increased efficiency.

But being analog and mechanically driven, it was prone to human and manufacturing errors which lead to incorrect inconsistent results.

This android app StopWatch eliminates the manufacturing error and enhances the user experience by realising the stopwatch to indeed an android application which is actually a computer code.

This will give the users a capability to measure time more precisely and hence be an aid in tasks that require efficiency to the fraction of seconds.

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the Minor Project entitled "**Stopwatch"** in partial fulfilment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering affiliated to **Guru Gobind Singh Indraprastha University, New Delhi** and submitted to the Department of Computer Science and Engineering ,Govind Ballabh Pant Govt. Engineering College, Okhla, is an authentic record of my own work

Carried out during a period from **June, 2018 to July 2018**. The matter represented in this report has not been submitted by me for award of any other degree of this or any other institute/university.

Date: - **Name**

Roll no.

This is to certify that the above statement made by the candidate is correct to the best of our knowledge.

Date

Signature of Supervisor

TABLE OF CONTENTS

INTRODUCTION

STOPWATCHES:-

Stopwatches can be classified into two categories, Type I and Type II. In general, stopwatches are classified as :-

Type I :-

if they have a digital design employing quartz oscillators and electronic circuitry to measure time intervals (Figure 2).

Type II:-

stopwatches have an analog design and use mechanical mechanisms to measure time intervals. Key elements of Type I and Type II stopwatches are summarised in Table.

Figure 2. Type I digital stopwatch.

Figure 3. Type II mechanical stopwatch.

Table 1:- Types of Stopwatches

Description	Type I Stopwatch	Type II Stopwatch
Operating Principle	Time measured by division of time base oscillator	Time measured by mechanical movement
Time Base	• Quartz oscillator	• Mechanical mainspring • Synchronous motor, electrically driven
Case	• Corrosion-resistant metal • Impact-resistant plastic	
Crystal	• Protects display • Allows for proper viewing • May be tinted • May employ magnification	• Protects dial/hands • Allows for proper viewing • Must be clear and untinted
Minimum Time Interval	• 48 h without replacement of battery	• 3 h without rewinding
Start and Stop	• Single control to start/stop • Audible signal of start/stop	
Reset	• Must reset stopwatch to zero	
Split Time (if equipped)	• Must indicate whether display mode is regular or split time.	
Force to Operate Controls	• Must not exceed 1.8 N (0.4046 lbf)	
Dial and Hands		• Face must be white • Graduations must be black or red • Hands must be black or red
Required Markings	• Unique, nondetachable serial number • Manufacturer's name or trademark • Model number (Type I only)	
Digital Display	• Provide delimiting character for hours, minutes, seconds (usually colon)	
Minimum	• 0.2 s	

BASIC THEORY OF OPERATION

Every stopwatch is composed of four elements: a power source, a time base, a counter, and an indicator or display. The design and construction of each component depends upon the type of stopwatch. Digital (Type I) Stopwatches — The power source of a type I stopwatch is usually a silver cell or alkaline battery, which powers the oscillator, counting and display circuitry. The time base is usually a quartz crystal oscillator, with a nominal frequency of 32 768 Hz (215 Hz). Figure 4 shows the inside of a typical device, with the printed circuit board, quartz crystal oscillator, and battery visible. The counter circuit consists of digital dividers that count the time base oscillations for the period that is initiated by the start/stop buttons. The display typically has seven or eight digits.

TIMERS

Timers, unlike stopwatches, count down from a preset time period instead of counting up from zero. They can be small, battery-operated devices that are used to signal when a certain time period has elapsed, or they can be larger devices that plug into a wall outlet and control other items. A parking meter is an example of a countdown timer. Inserting a coin starts the internal timer counting down from an initial preset point. When the time has elapsed, the "EXPIRED" flag is raised. One type of timer used extensively in industry is the process control timer. As their name implies, these devices measure or control the duration of a specific process. For example, when a product is made, it may need to be heat treated for a specific length of time. In an automated manufacturing system, the process control timer determines the amount of time that the item is heated. In some applications, such as integrated circuit manufacturing, the timing process can be critical for proper operation.

Process control timers are also used in many different types of laboratory environments. Calibration laboratories use timers to calibrate units such as radiation detectors, where they regulate the amount of time the detector is exposed to the radiation source. Any uncertainty in the time of exposure directly influences the uncertainty of the detector calibration. Timers are also used in the medical field. For example, medical laboratories use process control timers when specimen cultures are grown. Hospitals use timers to regulate the amount of medication given to patients intravenously.

ABSOLUTE ACCURACY SPECIFICATIONS:

The absolute accuracy2 of an instrument is the maximum allowable offset from nominal. Absolute accuracy is defined in either the same units or a fractional unit quantity of the measurement function for an instrument. For example, the absolute accuracy of a ruler might be specified as ±1 mm for a scale of 0 to 15 cm. In the case of timing devices, it isn't useful to provide an absolute accuracy specification by itself. This is because a device's time offset from nominal will increase as a function of time. If the timing device were able to measure an infinite time interval, the offset (or difference in time from nominal) of the device would also become infinitely large. Because of this, when timing devices are specified with an absolute accuracy number, it is also accompanied by a time interval for which this specification is valid. An example of this is the specifications for the stopwatch shown in Figure below, specified with an absolute accuracy of 5 s per day. If the stopwatch in Figure below were used to measure a longer time interval, we could determine a new absolute accuracy figure by simply multiplying the original specification by the desired time interval. For example, 5 s per day becomes 10 s per two days, 35 s per week, and so on.

While it is usually acceptable to multiply the absolute accuracy by time intervals longer than the period listed in the specifications, we must use caution when dividing the absolute accuracy specification for periods of time shorter than the period listed in the specifications. If we divide the absolute accuracy specification for shorter measurement periods, a new source of uncertainty, the resolution uncertainty of the instrument, becomes important to consider. For example, if we try to determine the accuracy of the stopwatch of Figure below for a period of 30 s, we can compute the absolute accuracy as follows:

$$\boxed{\frac{5\,\text{s}}{\text{day}} \times 30\,\text{s} = \frac{5\,\text{s}}{\text{day}} \times \frac{1}{2880}\,\text{day} = 0.0017\,\text{s}}$$

- Handsome stopwatch with large display provides timing to 1/100th of a second over a range of 9 hours 59 minutes and 59.99 seconds.
- Accurate to ± 5 s/day.
- Built-in memory recalls up to ten laps.
- Clock function (12 or 24 hour) features a programmable alarm with an hourly chime plus built-in calendar displays day, month and date.
- Countdown timer function features input ranges from one minute to 9 hours, 59 minutes.
- Dimensions/Weight: 2.5×3.2×.8 in (63×81×20 mm); 2.8 oz.
- Water resistant housing is complete with lithium battery.

A Digital Stopwatch used now.

Chapter 2
Software Description

The purpose of this document is to present a detailed description of the android application Stopwatch. It will explain the importance and features of the application, the interfaces of the application, what the application will do, the constraints under which it must operate .

6.2 Document Conventions

Main Section Titles

- Font: Times New Roman
- Face: Regular
- Size: 14 Sub Section Titles

Sub Section Titles

- Font: Times New Roman
- Face: Bold
- Size: 13

Other Text Explanations

- Font: Times New Roman
- Face: Normal
- Size: 12

Intended Audience and Reading Suggestions :-

This document is intended for students, developers, project manager, users, testers, documentation writers etc.
This document is organised as follows:

Product Scope

This android application will be a substitute fora stopwatch, and will offer services to the mobile user without any issues of compatibility or capability.

Overall Description

1:-Product Perspective

The StopWatch application will be a new Android based offline application. It will be implemented just like any other app

2:- Product Functions

The product has an easy to use interface which is a single screen view, comprising of start pause and reset function

3:- User Classes and Characteristics

Our application hopes to draw on three main user groups. The students almost all of them have smartphones. About half of those smartphones are Android, and more than half of those Android phones are fit to run Stopwatch, in the current version.

These users are obsessed with social media, and almost all of them actively uses at least one of the following: Facebook, Twitter, Instagram, Snapchat.

4:- Operating Environment

The application will only be available for the Android operating systems. The application shall only be used with compatible android devices The user shall use this application on Android OS 7.0 or any later versions of the Android OS.

Functional Requirements:-

This application requires an android smartphone with Android 7.0

Other Non Functional Requirements:-

PerformanceRequirements

Real-Time

The application will provide up-to-date information. It should display the latest results at all times, and if it lags behind, the user should be notified.

System Resource Consumption

Resource consumption of this application should not reach an amount that renders the mobile device unusable. The application should be capable of operating in the background should the user wish to utilize other applications.

<u>Software Quality Attributes</u>

Reliability

The application will meet all of the functional requirements without any unexpected behaviour. At no time should the gauge output display incorrect or outdated information without alerting the user to potential errors.

Availability

The application will be available at all times on the user‟s Android device, as long as the device is in proper working order. The functionality of the application will depend on any external services such as internet access that are required. If those services are unavailable, the user should be alerted.

Security

The software should never disclose any personal information of any users, and should collect no personal information from its own users.

Maintainability

The application can maintained easily .

Portability

This software will be designed to run on any Android operating system version 7.0 or higher. The software will be forward compatible for all currently released Android operating system versions beyond 7.0.

CHAPTER 3

Source code:-

JAVA CODE:-

```
package com.renitus.tusharpandey.stopwatch;


import android.os.Handler;
    import android.os.SystemClock;
    import android.support.v7.app.AppCompatActivity;
    import android.os.Bundle;
    import android.view.View;
    import android.widget.Button;
    import android.widget.TextView;


public class MainActivity extends AppCompatActivity {
  TextView timer ;
  Button start, pause, reset;
  long MillisecondTime, StartTime, TimeBuff, UpdateTime = 0L ;
  Handler handler;
  int Seconds, Minutes, MilliSeconds ;



  @Override
  protected void onCreate(Bundle savedInstanceState) {
```

```
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
timer = (TextView) findViewById(R.id.tvTimer);
start = (Button) findViewById(R.id.btStart);
pause = (Button) findViewById(R.id.btPause);
reset = (Button) findViewById(R.id.btReset);
handler = new Handler() ;

start.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View view) {

      StartTime = SystemClock.uptimeMillis();
      handler.postDelayed(runnable, 0);

      reset.setEnabled(false);

   }
});

pause.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View view) {
```

```
        TimeBuff += MillisecondTime;


        handler.removeCallbacks(runnable);


        reset.setEnabled(true);


    }
});


reset.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {


        MillisecondTime = 0L ;
        StartTime = 0L ;
        TimeBuff = 0L ;
        UpdateTime = 0L ;
        Seconds = 0 ;
        Minutes = 0 ;
        MilliSeconds = 0 ;


        timer.setText("00:00:00");


    }
```

```
    });

}

public Runnable runnable = new Runnable() {

    public void run() {

        MillisecondTime = SystemClock.uptimeMillis() - StartTime;

        UpdateTime = TimeBuff + MillisecondTime;

        Seconds = (int) (UpdateTime / 1000);

        Minutes = Seconds / 60;

        Seconds = Seconds % 60;

        MilliSeconds = (int) (UpdateTime % 1000);
        String st = Minutes + ":"
                +  Seconds + ":"
                +  MilliSeconds;
        timer.setText(st);
```

```
            handler.postDelayed(this, 0);
        }

    };

}
```

XML CODE:-

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

<android.support.constraint.ConstraintLayout 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="#00008b"

    tools:context="com.renitus.tusharpandey.stopwatch.MainActivity">




    <RelativeLayout

        android:layout_width="fill_parent"

        android:layout_height="fill_parent"

        android:layout_marginLeft="10dp"

        android:layout_marginRight="10dp"

        android:paddingBottom="90dp">



        <TextView

            android:text="0:0:0"

            android:layout_width="wrap_content"

            android:layout_height="wrap_content"
```

```
        android:id="@+id/tvTimer"
        android:textSize="50dp"
        android:textStyle="bold"
        android:textColor="#ffffff"
        android:layout_marginTop="120dp"
        android:paddingBottom="50dp"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true" />

    <Button
        android:text="Start"
        android:background="#ffffff"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/tvTimer"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginTop="41dp"
        android:id="@+id/btStart" />

    <Button
        android:text="Pause"
        android:background="#ffffff"
        android:layout_width="wrap_content"
```

```xml
        android:layout_height="wrap_content"
        android:id="@+id/btPause"
        android:layout_alignBaseline="@+id/btStart"
        android:layout_alignBottom="@+id/btStart"
        android:layout_centerHorizontal="true" />

    <Button
        android:text="Reset"
        android:background="#ffffff"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignTop="@+id/btPause"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:id="@+id/btReset" />

  </RelativeLayout>
</android.support.constraint.ConstraintLayout>
```

Source code explanation:-

Following text shall describe and elaborate the definition and throw light on the significance and working of various parts of the code used above:-

The application is divided into 2 parts:-

1. Java coding
2. XML coding

Java Coding:-

Java is a commonly used language and many programmers know it, it can run on a virtual machine (VM) so no need to recompile for different phones, better security, many development tools available for Java, and Java is a known industry language with most phones compatible with it.
Though Google provides the Java API, Android does not use JVM to execute class files. Rather, it uses Dalvik Virtual Machine (DVM). The class files are compiled into Dalvik Executable (DEX) format, and bundled as Android Package (APK) along with other resources.
With Java, if you are aware of object-oriented programming principles, creating applications for android will be much simpler than iOS app development.

In the code:-

the java coding provides a structure of dataflow, how the code executes and hence how the application will behave.

This part has little to do with the UI of the app but has emphasis on the working and interaction of the UI and hence the user to the mobile phone

XML Coding:-

XML stands for **eXtensible Markup Language.**
It is used for ‘drawing’ the interfaces of an application.
JAVA is used for writing the backend (developer's end) codes while frontend (user's end) codes are written on XML.
A program code has no value without a good layout and design.

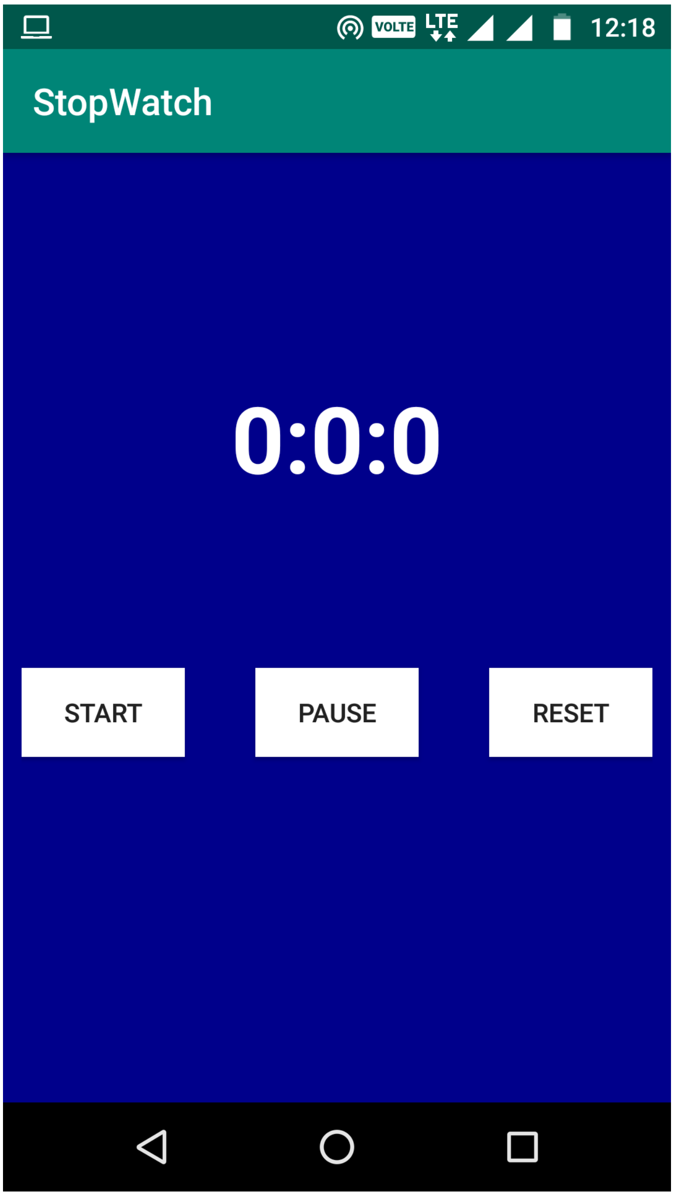
In Android XML is used to define layouts, colors, fonts, styles, strings, arrays, shapes and app configuration manifest file. You can define your app configuration in xml and read it from or data from remote service can be in XML format if your data has complex structure.

In the code:-

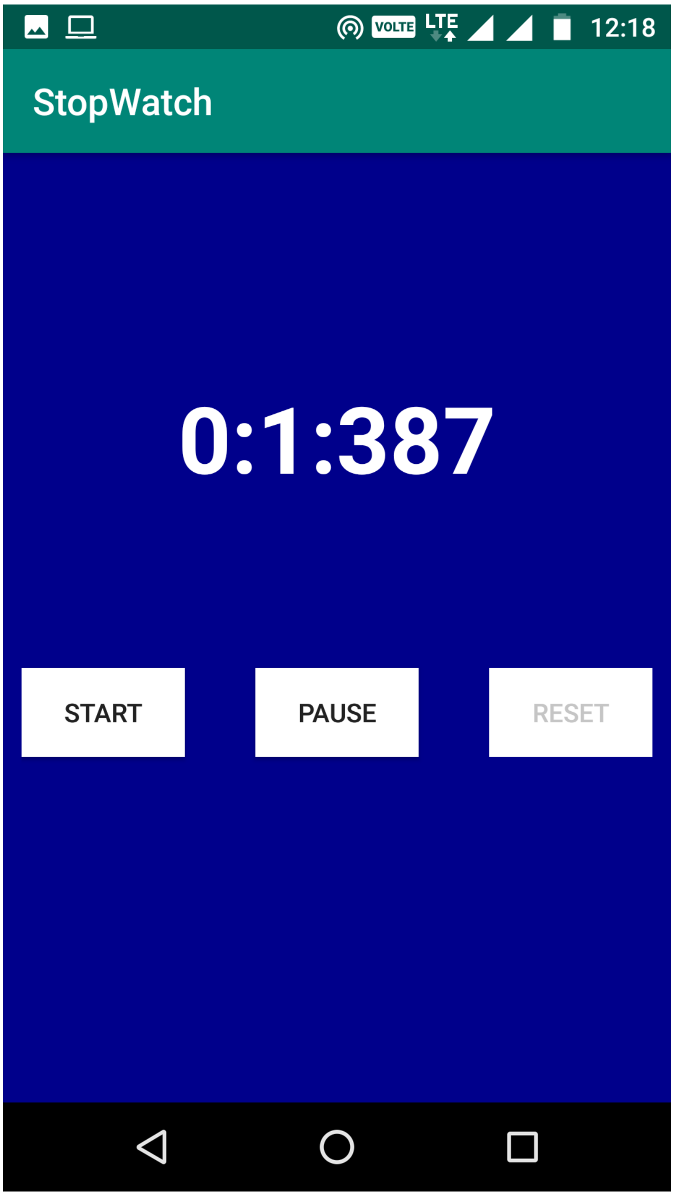
the XML coding is the part that defines and helps render the UI of the whole application.

This is the part that interacts with the user and hence is crucial to be designed and encode in a way to give a lucid visible interface that is easy to interact with and hence to be used.

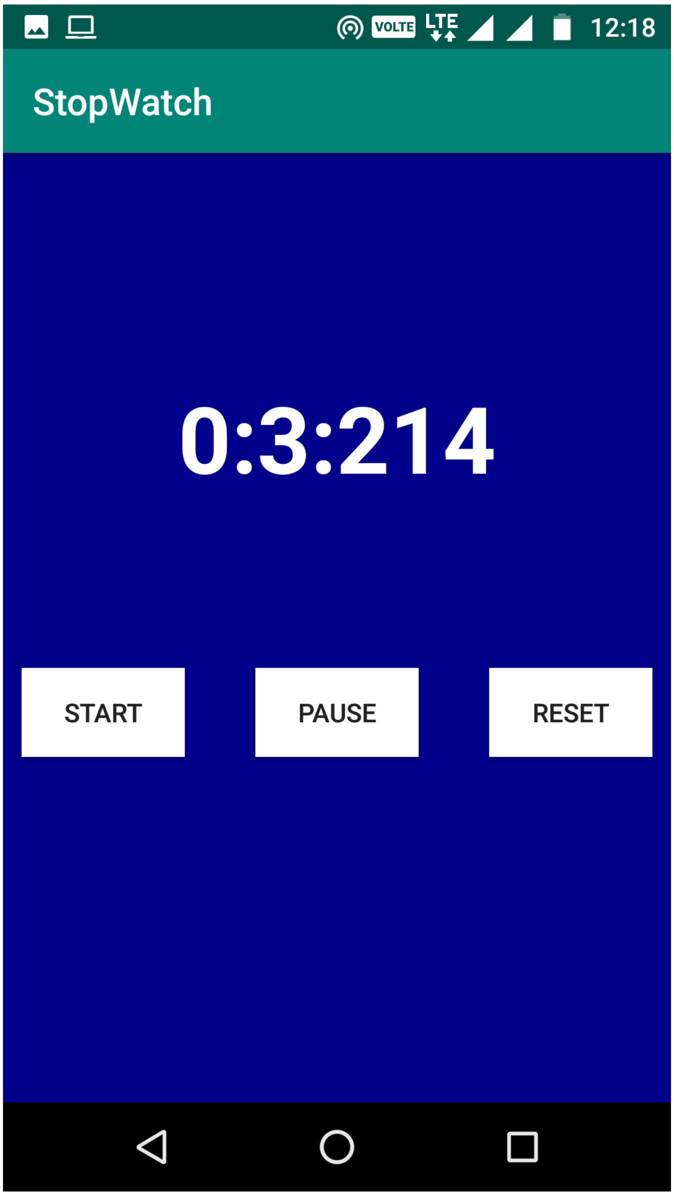
OUTPUT SCREENSHOTS:-



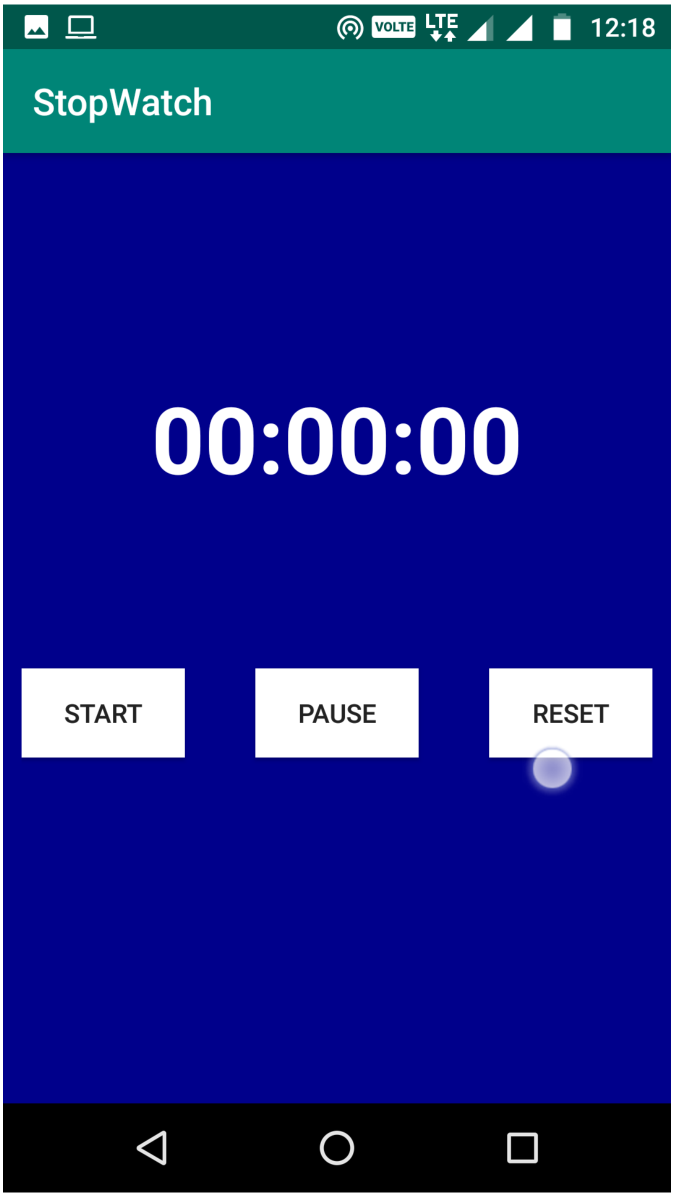
The welcoming first screen



The start button is pressed and stopwatch started



The Screenshot of moment when stopWatch was paused by pause button



Reset button pressed and stopwatch timer reset

CHAPTER 5

APPLICATION

CHAPTER 5

5.1 APPLICATIONS.

- Laboratory experiments
- Sporting events
- Competitions
- Commercial Environments
- HouseHold Chores, or even Family Games

CONCLUSION

After completion of this android stopwatch application, I have learnt some important information about the android operating systems and various aspects of the Java, XML, and Android Studio-the IDE for android application development.

REFERENCES

1.) stackoverflow.com - theoretical knowledge on android function and code used

2.) Google Images, **pinterest.au**, - all images in the report

3.) **tutorialspoint.com** - information on Application structure.