TO CREATE TIMER CLOCK

AIM:

To create timer clock using Mobile Application.

ALGORITHM:

Step 1: Create the activity_main.xml layout file with a TextView for displaying the time and three buttons for starting, stopping, and resetting the stopwatch.

Step 2: Define the TextView to show the time in "00:00:00" format with a large font size. Place it in the center of the screen.

Step 3: Add the startButton, stopButton, and resetButton within a horizontal LinearLayout, placed below the TextView for user interaction.

Step 4: The MainActivity.java, initialize the TextView and buttons using findViewById() to link them with the XML layout components.

Step 5: Implement the startStopwatch() method to start tracking time. Capture the current time using System.currentTimeMillis() and update the time display every second using a Handler and a Runnable to run the code repeatedly.

Step 6: Implement the stopStopwatch() method to pause the stopwatch. This method removes the callback from the Handler, effectively stopping the time updates.

Step 7: Implement the resetStopwatch() method to reset the displayed time back to "00:00:00" and stop the stopwatch if it is running.

Step 8: Define the main activity in AndroidManifest.xml and ensure it launches by adding the required intent filter. The stopwatch does not require any special permissions.

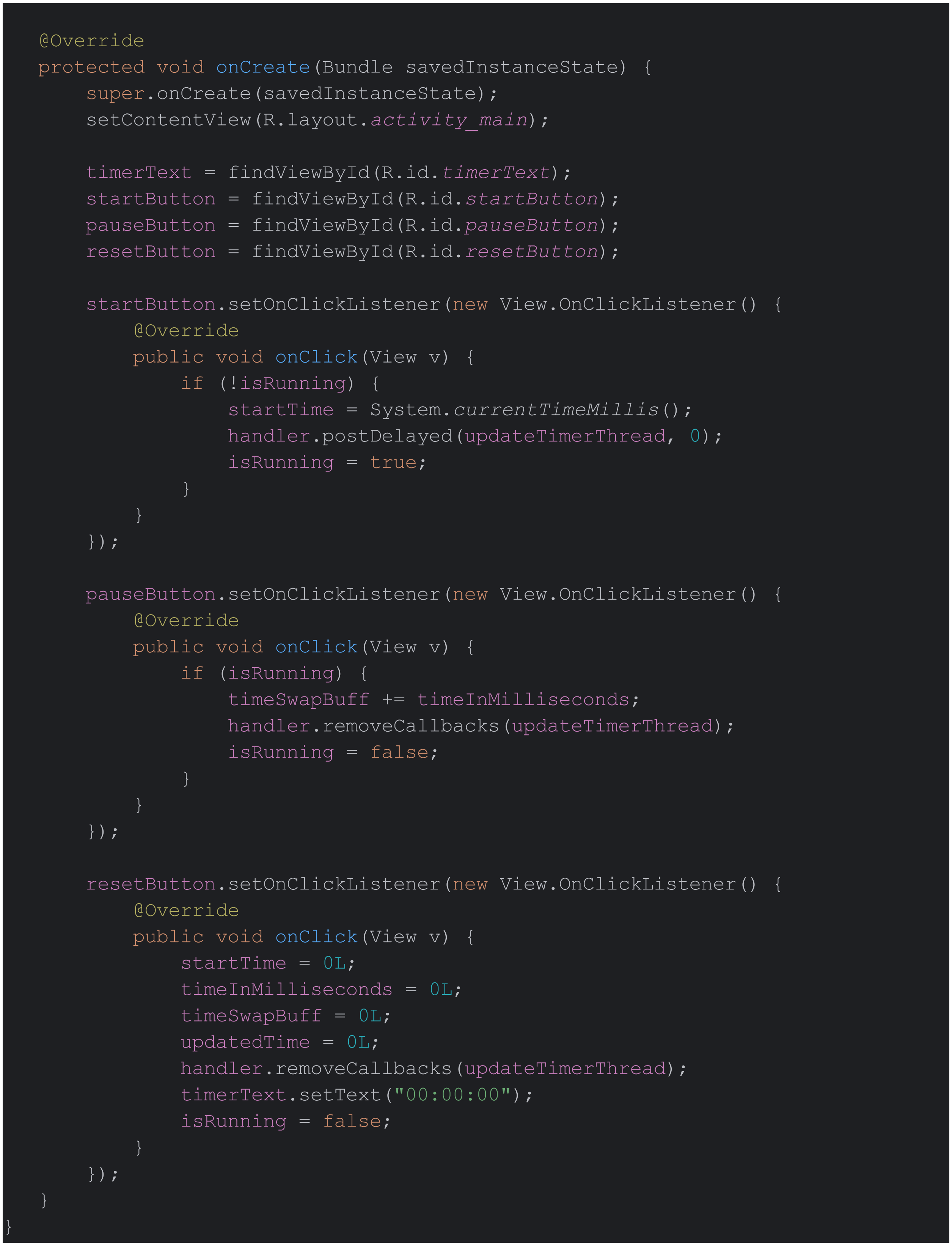
Step 9: Stop the Process.

ACTIVITY_MAIN.XML

```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    android:background="#F5F5F5">

    <TextView
        android:id="@+id/timerText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="50dp"
        android:text="00:00:00"
        android:textSize="48sp"
        android:textColor="#000000"
        android:fontFamily="sans-serif-bold"/>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/timerText"
        android:layout_marginTop="40dp"
        android:orientation="horizontal"
        android:gravity="center">

        <Button
            android:id="@+id/startButton"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Start"
            android:backgroundTint="#4CAF50"
            android:textColor="#FFFFFF"
            android:layout_marginEnd="10dp"/>

        <Button
            android:id="@+id/pauseButton"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Pause"
            android:backgroundTint="#FFC107"
            android:textColor="#FFFFFF"
            android:layout_marginEnd="10dp"/>

        <Button
            android:id="@+id/resetButton"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Reset"
            android:backgroundTint="#F44336"
            android:textColor="#FFFFFF"/>

    </LinearLayout>

</RelativeLayout>
```

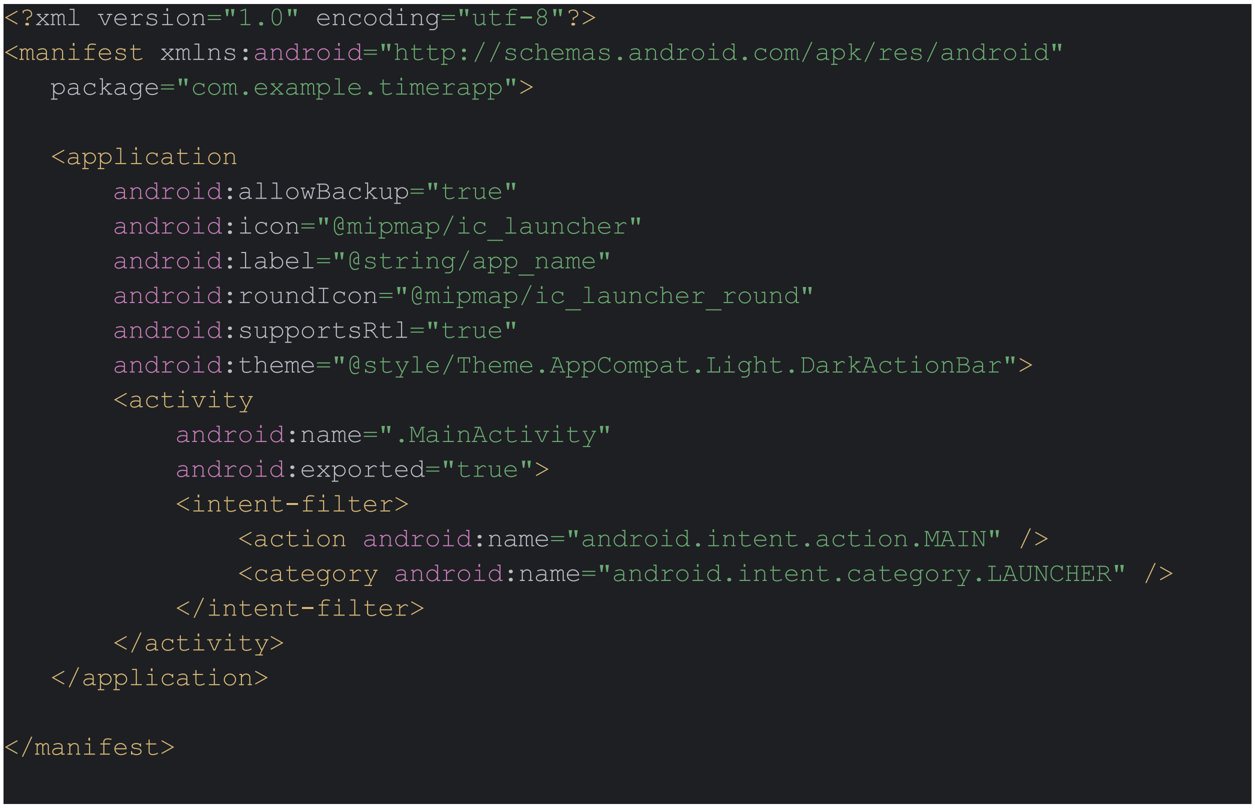
MAINACTIVITY.JAVA

```
package com.example.timerapp;

import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private TextView timerText;
    private Button startButton, pauseButton, resetButton;
    private Handler handler = new Handler();
    private long startTime = 0L;
    private long timeInMilliseconds = 0L;
    private long timeSwapBuff = 0L;
    private long updatedTime = 0L;
    private boolean isRunning = false;

    private Runnable updateTimerThread = new Runnable() {
        public void run() {
            timeInMilliseconds = System.currentTimeMillis() - startTime;
            updatedTime = timeSwapBuff + timeInMilliseconds;

            int secs = (int) (updatedTime / 1000);
            int mins = secs / 60;
            int hrs = mins / 60;
            secs = secs % 60;
            mins = mins % 60;

            timerText.setText(String.format("%02d:%02d:%02d", hrs, mins, secs));
            handler.postDelayed(this, 1000);
        }
    };
```



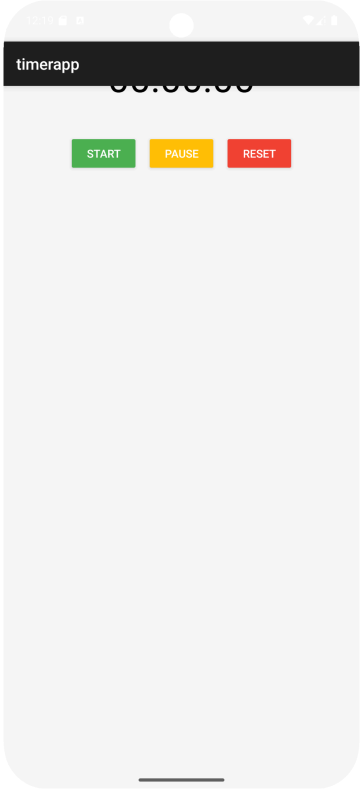
```java
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        timerText = findViewById(R.id.timerText);
        startButton = findViewById(R.id.startButton);
        pauseButton = findViewById(R.id.pauseButton);
        resetButton = findViewById(R.id.resetButton);

        startButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (!isRunning) {
                    startTime = System.currentTimeMillis();
                    handler.postDelayed(updateTimerThread, 0);
                    isRunning = true;
                }
            }
        });

        pauseButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (isRunning) {
                    timeSwapBuff += timeInMilliseconds;
                    handler.removeCallbacks(updateTimerThread);
                    isRunning = false;
                }
            }
        });

        resetButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startTime = 0L;
                timeInMilliseconds = 0L;
                timeSwapBuff = 0L;
                updatedTime = 0L;
                handler.removeCallbacks(updateTimerThread);
                timerText.setText("00:00:00");
                isRunning = false;
            }
        });
    }
}
```

ANDROIDMAINFEST.XML



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="com.example.timerapp">

   <application
       android:allowBackup="true"
       android:icon="@mipmap/ic_launcher"
       android:label="@string/app_name"
       android:roundIcon="@mipmap/ic_launcher_round"
       android:supportsRtl="true"
       android:theme="@style/Theme.AppCompat.Light.DarkActionBar">
       <activity
           android:name=".MainActivity"
           android:exported="true">
           <intent-filter>
               <action android:name="android.intent.action.MAIN" />
               <category android:name="android.intent.category.LAUNCHER" />
           </intent-filter>
       </activity>
   </application>

</manifest>
```

OUTPUT


timerapp
START
PAUSE
RESET