VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belgaum-590018.

MOBILE APPLICATION DEVELOPMENT
(18CSMP68)

MINI PROJECT REPORT

on

"Asynchronous Banner Task with Improvised Background Video"

Submitted in partial fulfillment for the requirements of the VI Semester degree of

BACHELOR OFENGINEERING IN
COMPUTER SCIENCE AND ENGINEERING

For The Academic Year
2021-22
By

NEHA R RAO (1DB19CS095)
RUCHITHA M (1DB19CS165)
LATHARANI (1DB20CS405)

Under The Guidance of

Prof. PRAKRUTHI S T
Assistant Professor,
Dept. of CSE

Department of Computer Science and Engineering
DON BOSCO INSTITUTE OF TECHNOLOGY
Kumbalagodu, Mysore Road, Bengaluru - 560 074.

DON BOSCO INSTITUTE OF TECHNOLOGY

Kumbalagodu, Bengaluru – 560 074.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the mini project report entitled **"Asynchronous Task with Improvised Background Video"** is a bonafide work carried out by **Neha R Rao (1DB19CS095), Ruchitha M (1DB19CS165) and Latharani (1DB20CS405)** in partial fulfillment of award of Degree of **Bachelor of Engineering in Computer Science and Engineering** of Visvesvaraya Technological University, Belagavi, during the academic year 2021- 2022. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated. The mini project has been approved as it satisfies the academic requirements associated with the degree mentioned.

Signature of guide

……

Prof. Prakruthi S T
Assistant Professor
Dept. of CSE,
DBIT, Bengaluru.

Signature of HOD

…….

Dr. K B Shivakumar
Head of Dept.,
Dept. of CSE,
DBIT, Bengaluru.

DON BOSCO INSTITUTE OF TECHNOLOGY

Kumbalagodu, Bangalore – 560 074.

DECLARATION

We, **Neha R Rao, Ruchitha M and Latharani** students of sixth semester B.E, Department of Computer Science and Engineering, Don Bosco Institute of Technology, Kumbalagodu, Bangalore, declare that the miniproject work entitled **"Asynchronous Task with Improvised Background Video"** has been carried out by us and submitted in partial fulfillment of the course requirements for the award of degree in **Bachelor of Engineering** in **Computer Science and Engineering** of **Visvesvaraya Technological University, Belgaum** during the academic year **2021-22** . The matter embodied in this report has not been submitted to any other university or institution for the award of any other degree or diploma.

Place: Bangalore
Date: 14-07-2022

Neha R Rao(1DB19CS095)
Ruchitha M(1DB19CS165)
Latharani (1DB20CS405)

ACKNOWLEDGEMENT

We're here by submitting Mobile Application Development Mini Project entitled **"Asynchronous Task with Improvised Background Video"** as per the scheme of Visvesvaraya Technological University, Belgaum. In this connection, we would like to express our deep sense of gratitude to our beloved institution Don Bosco Institute of Engineering and also, we would like to express our sincere gratitude and indebtedness to **Prof. B.S. Umashankar, Principal, DBIT, Bengaluru.**

We would like to express our sincere gratitude to **Dr. K.B. Shivakumar, Head of Dept. of Computer Science and Engineering, DBIT** for providing a congenial environment to work in and carryout our mini project.

We would like to express the deepest sense of gratitude to thank our Project Guide
Prof. Prakruthi S T, Assistant Professor, Dept. of Computer Science and Engineering, DBIT for her constant help and support extended towards us during the course of the project.

Finally, we are very much thankful to all the teaching and non-teaching members of the Department of Computer Science and Engineering, my seniors, friends and my parents for their constant encouragement, support and help throughout completion of report.

NEHA R RAO (1DB19CS095)
RUCHITHA M (1DB19CS165)
LATHARANI (1DB20CS405)

ABSTRACT

The main aim of this project is to demonstrate the use of Asynchronous task in android AsyncTask which is designed to be a helper class around thread and handler and does not constitute a generic threading framework. AsyncTasks are to be ideally used for short operations (a few seconds at the most). An asynchronous task is defined by a computation that runs on a background thread and the result is published on the UI thread.

Implementation of Asynchronous task using the functionality of a simple moving banner with a set of start and stop buttons and improvise it by including an additional feature like background video using a view flipperclass which displays toast messages.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

In recent years, the emergence of smart phones has changed the definition of mobile phones. Phone is no longer just a communication tool, but also an essential part of the people's communication and daily life. Various applications adds unlimited fun for people's lives. It is certain that the future of the network will be the mobile terminal. Now the Android system in the electronics market is becoming more and more popular, especially in the smartphone market. Because of the open source, some of the development tools are free, so there are plenty of applications generated. This has greatly inspired the people to use the Android system. In addition, it provides a very convenient hardware platform for developers so that they can spend less effort to realize their ideas.

After studying Android applications and going through few resources, we have utilized Java language, Android studio, Android ADT and the Android SDK to develop the mobile application. These systems have a nice interface, smooth operation and provide a wonderful user experience.

Android

Android is a mobile operating system (OS) currently developed by Google, based on the Linux kernel and designed primarily for touchscreen mobile devices such as smartphones and tablets. Android's user interface is mainly based on direct manipulation, using touch gestures that loosely correspond to real-world actions, such as swiping, tapping and pinching, manipulating on-screenobjects along with a virtual keyboard for text input. In addition to touchscreen devices, Google has further developed Android TV for televisions, Android Auto for cars, and Android Wear for wrist watches which are specialized with user interface. Variants of Android are also used on notebooks, game consoles, digital cameras, and other electronics. Initially developed by Android, Inc., which Google bought in 2005, Android was unveiled in 2007, along with the founding of the Open Handset Alliance – a consortium of hardware, software, and telecommunication companies devoted to advancing open standards for mobile devices. As of July 2013, the Google Play store has had over one million Android applications ("apps") published, and over 50 billion applications downloaded. An April–May 2013 survey of mobile application developers found that 71% of developers

create applications for Android, and a 2015 survey found that 40% of full-time professional developers see Android as their priority target platform.

Software Development Kit (SDK)

A software development kit (SDK or "devkit") is typically a set of software development tools that allows the creation of applications for a certain software package, software framework, hardware platform, computer system, video game console, operating system, or similar development platform. To create applications, downloading the software development kit is necessary. For example, to create an Android app SDK with java programming is required, for iOS apps iOS SDK with swift language is required and to develop MS Windows apps with the.net language is required. There are also SDKs that are installed in apps to provide analytics and data about activity. Prominent examples include Google and Facebook.

Android Studio

Android Studio is an integrated development environment (IDE) for developing the Android platform and was announced on May 16, 2013 at the Google I/O conference. Android Studio is freely available under the Apache License 2.0. Android Studio which 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 to currently Arctic Fox Based on JetBrains' IntelliJ IDEA software, Android Studio is designed specifically for Android development. It is available for download on Windows, Mac OS X and Linux and has replaced Eclipse Android Development Tools (ADT) as Google's primary IDE for native Android application development.

CHAPTER 2

PROBLEM STATEMENT AND OBJECTIVES

2.1 PROBLEM STATEMENT:

Develop an application to demonstrate the use of Asynchronous tasks in android. The asynchronous task should implement the functionality of a simple moving banner. On pressing the **Start Task** button, the banner message should scroll from right to left. On pressing the **Stop Task** button, the banner message should stop. Let the banner message be "Demonstration of Asynchronous Task with improvised background video".

2.2 OBJECTIVES:

1. To design an xml code that displays a moving banner of messages using the asynchronous task class and implementing it using java
2. Adding additional features like background video to improve system and user interface.
3. Testing the code for all its intended functionalities using an android device or AVD.

CHAPTER 3

LITERATURE SURVEY

The Android - A Widely Growing Mobile Operating System with its Mobile based application has been in extreme demand for developing software that runs on mobile devices. The developers have to keep up with this high demand and deliver high-quality app on time and within budget. For this,estimation of development and testing of apps play a pivotal role.

Android Studio is an integrated development environment (IDE) for developing Android applications and was announced on May 16, 2013, at the Google I/O conference. The first stable build was released in December 2014, starting from version 1.0 to currently Arctic Fox

Few apps like 1. LED Scroller. (A Very Simple Electronic Display App) does display a moving banner but is of the conventional fixed led type and fails to displayadvanced graphical backgrounds and fonts.

CHAPTER 4

PROPOSED SYSTEM

The proposed system is meant to carry out the demonstration of an "Asynchronous Task" class with the help of a moving banner and incorporation of additional functionalities like background video.

An asynchronous class which is used to communicate with the background threads and processes is used to implement a moving banner, also a view flipper method is used to implement background video.

The proposed system is implemented in several stages:

- Designing an interface in xml.
- Coding for the implementation in Java.
- Addition of background video feature using viewflipper.

CHAPTER 5

ANALYSIS

5.1 : HARDWARE REQUIREMENTS:

- **PROCESSOR :** x86_64 CPU architecture; 2nd generation Intel Core or newer,or AMD CPU with support for a Windows Hypervisor.
- **RAM:** 8 GB RAM or more.
- **DISK SPACE:** 8 GB of available disk space minimum (IDE + Android SDK +Android Emulator).
- **DISPLAY:** 1280 x 800 minimum screen resolution.

5.2: SOFTWARE REQUIREMENTS:

- **OPERATING SYSTEM :** 64-bit Microsoft® Windows® 8/10
- **EMULATOR :** Pixel 4 API 30
- **TARGET SDK VERSION :** 30
- **SDK BUILD TOOLS VERSION :** 30.0.3
- **LANGUAGES USED:** XML , Java
- **JDK VERSION:** Java Development Kit 8

CHAPTER 6

APPLICATION DESIGN

- **TextView - A user interface element that displays text to the user.**
- **EditText - A user interface element that displays editable text to the user.**
- **ConstraintLayout - is a ViewGroup which allows you to position and size**
- **Button - A user interface element which the user can tap or click on to perform an action widget in a flexible way.**

6.1.1 XML CODE (activity main.xml) :

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
 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"
 tools:context=".MainActivity"
 android:padding="16dp">

 <LinearLayout
   android:id="@+id/linearLayout"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   tools:ignore="MissingConstraints">

<ViewFlipper
  android:id="@+id/bckgrndViewFlipper1"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent">

<FrameLayout
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:layout_below="@id/txtView">

<VideoView
  android:id="@+id/videoView"
  android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content" />

       </FrameLayout>
       </ViewFlipper>

   </LinearLayout>

   <TextView
      android:id="@+id/txtView"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginTop="350dp"
      android:background="#2196F3"
      android:fadingEdge="vertical"
      android:fontFamily="@font/alata"
      android:foregroundTint="#009688"
      android:scrollHorizontally="true"
      android:singleLine="true"
      android:text="Cherish the reefs while you can - Corals Are Already A Gift, Don't Give Them As
       Presents"
      android:textAppearance="@style/TextAppearance.AppCompat.Body2"
      android:textColor="@color/black"
      android:textSize="30sp"
      android:textStyle="bold"
      app:counterEnabled="true"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintHorizontal_bias="0.0"
      app:layout_constraintLeft_toLeftOf="parent"
      app:layout_constraintRight_toRightOf="parent"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="@+id/linearLayout" />

   <Button
      android:id="@+id/btnStart"
      android:layout_width="142dp"
      android:layout_height="71dp"
      android:layout_marginTop="115dp"
      android:layout_marginEnd="200dp"
      android:background="@color/material_on_primary_emphasis_medium"
      android:text="START"
      android:textAppearance="@style/TextAppearance.AppCompat.Large"
      android:textStyle="bold"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintHorizontal_bias="0.935"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toBottomOf="@+id/txtView" />
```

```
<Button
android:id="@+id/btnStop"
android:layout_width="135dp"
android:layout_height="71dp"
android:layout_marginStart="200dp"
android:layout_marginTop="115dp"
android:background="@color/material_on_primary_emphasis_medium"
android:backgroundTint="#00BCD4"
android:text="STOP"
android:textAppearance="@style/TextAppearance.AppCompat.Large"
android:textStyle="bold"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.072"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/txtView" />


</androidx.constraintlayout.widget.ConstraintLayout>
```

CHAPTER 7

IMPLEMENTATION

7.1 OVERVIEW OF SYSTEM IMPLEMENTATION:

7.1.1 USAGE OF SYNCHRONUS TASK CLASS:

AsyncTask is designed to be a helper class around Thread and Handler and does not constitute ageneric threading framework. AsyncTasks should ideally be used for short operations (a few seconds at the most.). To keep threads running for long period of time, it is highly recommended to use the various APIs provided by the java.util.concurrent package such as Executor, ThreadPoolExecutor and FutureTask.

An asynchronous task is defined by a computation that runs on a background thread and the result is published on the UI thread. An asynchronous task is defined by 3 generic types, called Params, Progress and Result, and 4 steps, called onPreExecute, doInBackground, onProgressUpdate and onPostExecute. When an asynchronous task is executed, the task goes through 4 steps:

- onPreExecute(), invoked on the UI thread before the task is executed. This step is normally used to setup the task, for instance by showing a progress bar in the userinterface.

- doInBackground(Params...), invoked on the background thread immediately after onPreExecute() finishes executing. This step is used to perform background computation that can take a long time. The parameters of the asynchronous task are passed to this step. The result of the computation must be returned by this step and willbe passed back to the last step. This step can also use publishProgress(Progress...) to publish one or more units of progress. These values are published on the UI thread, in the onProgressUpdate(Progress...) step.

- onProgressUpdate(Progress...), invoked on the UI thread after a call to publishProgress(Progress...). The timing of the execution is undefined. This method isused to display any form of progress in the user interface while the background computation is still executing. For instance, it can be used to animate a progress bar or show logs in a text.

- onPostExecute(Result), invoked on the UI thread after the background computation finishes. The result of the background computation is passed to this step as a parameter.

7.1.2 USAGE OF VIEW FLIPPER TO IMPLEMENT BACKGROUND VIDEO:

Simple ViewAnimator that will animate between two or more views that have been added to it.Only one child is shown at a time. If requested, can automatically flip between each child at a regular interval.

startFlipping:

Added in API level 1

public void startFlipping ()

Start a timer to cycle through child views

stopFlipping:

Added in API level 1

public void stopFlipping ()

No more flips

setFlipInterval:

Added in API level 1

public void setFlipInterval (int milliseconds)

How long to wait before flipping to the next vie

7.2 JAVA CODE (main activity.java):

```
package com.example.asynctask2;

import androidx.appcompat.app.AppCompatActivity;
import android.os.AsyncTask;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.Button;
import android.widget.MediaController;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.VideoView;
import android.widget.ViewFlipper;
public class MainActivity extends AppCompatActivity {
private Button btn1,btn2;
Animation fade_in,fade_out;
ViewFlipper viewFlipper;
TextView txt;
VideoView video;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    viewFlipper=(ViewFlipper)this.findViewById(R.id.bckgrndViewFlipper1);
    VideoView videoView = findViewById(R.id.videoView);
    videoView.setVideoPath("android.resource://"+getPackageName()+"/"+
    R.raw.v2);
    MediaController mediaController = new MediaController(this);
    mediaController.setAnchorView(videoView);
    videoView.setMediaController(mediaController);
    fade_in = AnimationUtils.loadAnimation(this, android.R.anim.fade_in);
    fade_out = AnimationUtils.loadAnimation(this, android.R.anim.fade_out);
    viewFlipper.setInAnimation(fade_in);
    viewFlipper.setOutAnimation(fade_out);
    //sets auto flipping
    viewFlipper.setAutoStart(true);
    viewFlipper.setFlipInterval(5000);
    viewFlipper.startFlipping();
    txt=findViewById(R.id.txtView);
```

```
    AsyncClass asyncClass=new AsyncClass();
    btn1=findViewById(R.id.btnStart);
    btn2=findViewById(R.id.btnStop);
    btn1.setOnClickListener(new View.OnClickListener() {
    @Override
      public void onClick(View v) {
                    asyncClass.doInBackground();
                    asyncClass.onProgressUpdate();
            }
      });
            btn2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                    txt.setSelected(false);
                    asyncClass.onPostExecute("LET US MAKE THE CORAL FLORAL AGAIN");
            }
      });
  }
  private class AsyncClass extends AsyncTask<String,String,String>{
    @Override
    protected void onPreExecute() {
      super.onPreExecute();
    }
    @Override
    protected void onProgressUpdate(String... values) {
      super.onProgressUpdate(values);
      Toast.makeText(getApplicationContext(),"BE MORAL WITH
      CORAL",Toast.LENGTH_SHORT).show();
    }
    @Override
    protected String doInBackground(String... strings) {
      txt.setEllipsize(TextUtils.TruncateAt.MARQUEE);
      txt.setSelected(true);
      return null;
    }
    @Override
    protected void onPostExecute(String s) {
      super.onPostExecute(s);
      Toast.makeText(getApplicationContext(),s,Toast.LENGTH_SHORT).show();
    }
  }
}
```

CHAPTER 8

RESULTS AND SCREENSHOTS

The results shown below is a representation of the mini project on android studio based on implementation of asynchronous task by using a moving banner design.

A TextView with horizontally moving enabled banner design should start moving from left to right on pressing a button and should stop moving on press of another mapped button and respective toast messages are displayed accordingly.

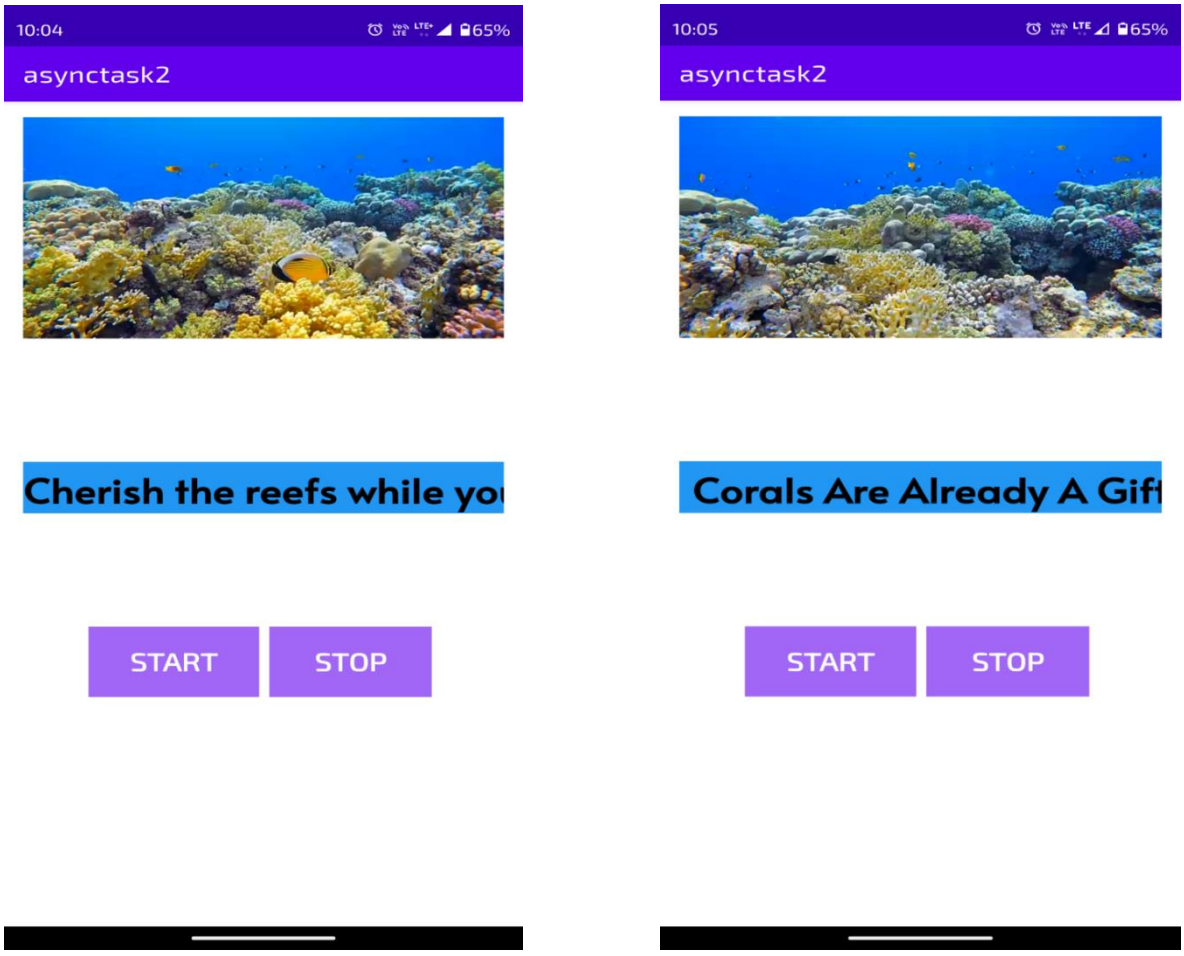

Fig 8.1 Demonstration of movement of banner upon button press and the toast messages associated

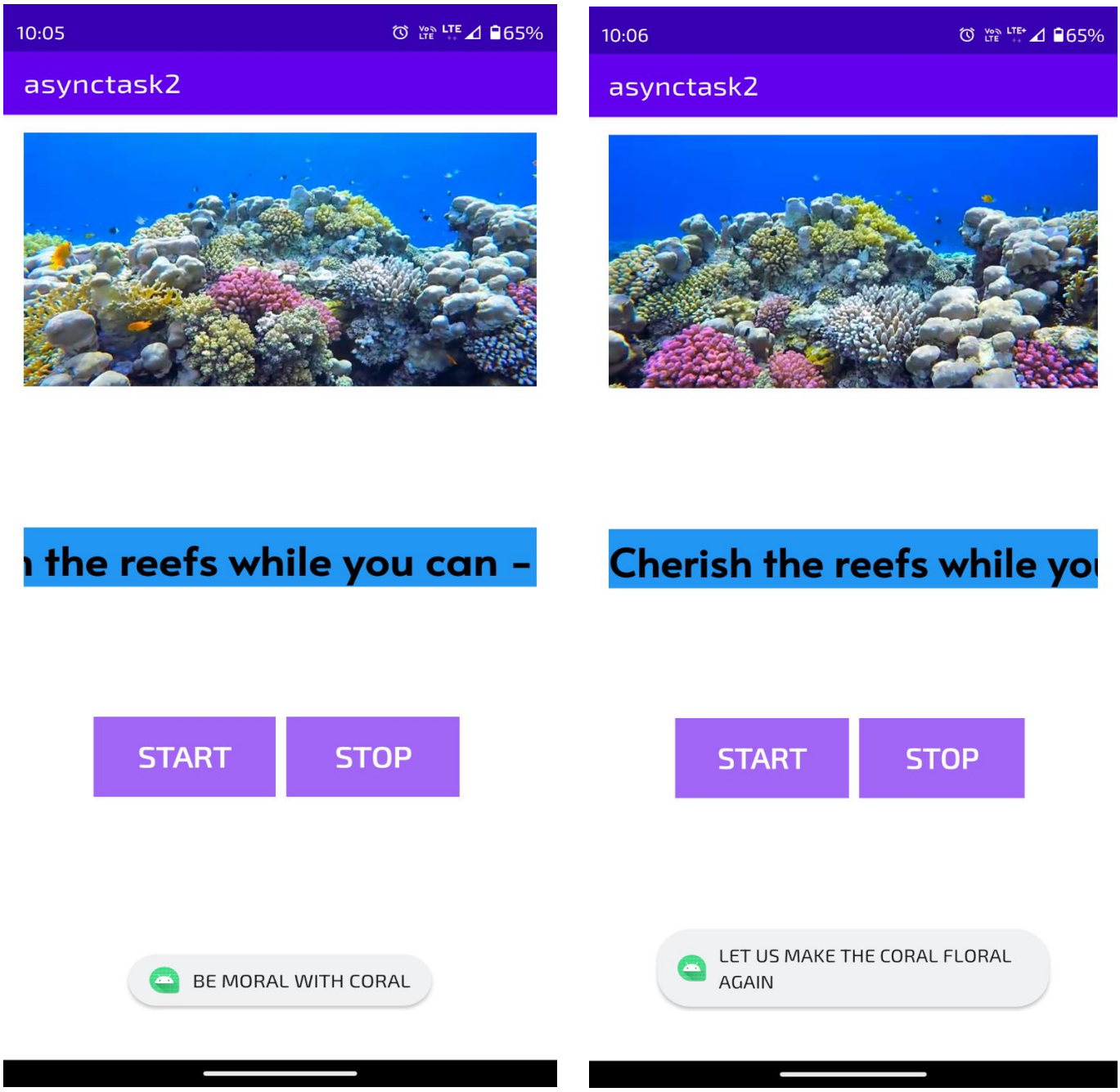

Fig 8.2 Demonstration of change in background after a certain period of time as a video implementation

CONCLUSION

Android as a full, open, and free mobile device platform, with its powerful function and good user experience rapidly has developed into the most popular mobile operating system. This report gives an overview of the different challenges and issues faced in android app development .The experienceof developing an android app is quite challenging, motivating as well as satisfying.

An asynchronous class which is used to communicate with the background threads and processes is used to implement a moving banner, also a view flipper method is used to implement background video.

FUTURE SCOPE

In ‘Digital Display’ boards the outdoor advertising can be changed quickly at a very low cost. Though the initial cost of purchasing and installing the **digital display board** will be higher than conventional painted or printed outdoor advertising, there are a number of advantages of using these electronic display boards. **Most outdoor advertising** is located at a height so that it can be visible to a large number of people in the surrounding area. The other advantage of placing it at a height is that it cannot be easily tampered. Hence it will be difficult for the advertiser or his advertising agency to reach the location and change the advertising message or display.

Digital user interface based banner design can be replaced with conventional fixed led bus hoardings which needs to be programmed to display sequence of LED’s to be lit up.

Future planned enhancements for the project include automatic change of background color based on time of the day to compensate for outdoor brightness.

BIBLIOGRAPHY

[1] Textbook: Google Developer Training, "Android Developer Fundamentals Course – Concept Reference", Google Developer Training Team, 2017

[2] https://github.com/1902shubh/SendMail

[3] https://www.gitbook.com/book/google-developer-training/android-developer-fundamentals-course- concepts/details

[4] https://developer.android.com/reference/android/os/AsyncTask

[5] https://developer.android.com/reference/android/widget/ViewFlipper