# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

**Jnana Sangama, Belagavi – 590018.**



**MINI PROJECT REPORT ON**

**“ASYNCHRONOUS TASK”**

Submitted in partial fulfillment for the requirement of VI semester for the

**Degree of Bachelor of Engineering in INFORMATION SCIENCE & ENGINEERING**

For the academic year 2022-23

**SUBMITTED BY:**

**DEEPIKA M L [1DB20IS043]**

**Under the guidance of:**

**Mr. ABHINAND B V Assistant Professor,**

**Dept. of ISE, DBIT**

**DON BOSCO INSTITUTE OF TECHNOLOGY, BENGALURU-560074**

**DON BOSCO INSTITUTE OF TECHNOLOGY**



**Kumbalagodu, Bengaluru-560074**

## DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING CERTIFICATE

This is to certify that the Mini Project Report entitled **“ASYNCHRONOUS TASK”** is a bonafide Mini Project work carried out by **DEEPIKA M L(1DB20IS043)**, in partial fulfillment of VI semester for the Degree of **Bachelor of Engineering in Information Science and Engineering** of Visvesvaraya Technological University, Belagavi, during the academic year 2022-23. It is certified that all corrections/suggestions indicated for Internal Assessments have been incorporated with the degree mentioned.

**Project Guide Head of Department**

**Mr. Abhinand B.V Dr. Raghavendra B.K**

Assistant Professor Professor & HoD

Dept. of ISE, Dept. of ISE,

DBIT, Bengaluru. DBIT, Bengaluru.

**External Viva**

**Name of the Examiners Signature with Date**

1.

*2.*

## DON BOSCO INSTITUTE OF TECHNOLOGY

**Kumbalagodu, Bengaluru -560074**



## DECLARATION

I **DEEPIKA M L**, student of sixth semester B.E, Department of Information Science and Engineering, Don Bosco Institute of Technology, Kumbalagodu, Bengaluru, declare, that Mini Project Work entitled **“ASYNCHRONOUS TASK”** has been carried out by and submitted in partial fulfillment of the requirement of VI semester. The matter embodied in this report has not been submitted to any university or institute for the award of any other degree or diploma.

**Place:** Bengaluru  **DEEPIKA M L**

**Date: (1DB20IS043)**

# ACKNOWLEDGEMENT

At the various stages in making the mini project, a number of people have given me invaluable comment on the manuscript. I take this opportunity to express my deepest gratitude and appreciation to all those who helped me directly or indirectly towards the successful completion of this mini project.

I would like to thank **Dr. Naghabhushana B.S, Principal, Don Bosco Institute of Technology** for his support though out this mini project.

I express my wholehearted gratitude to **Dr. B.K Raghavendra,** who is our respectable **Professor** and **Head of Department of Information Science and Engineering**. I wish to acknowledge for his valuable help and encouragement.

In this regard, a heartfelt gratitude to guide **Mr. Abhinand B.V, Assistant Professor of Department of Information Science and Engineering**, for his timely advice on the mini project and regular assistance throughout the mini project work. I would also like to thank the staff members of **Department of Information Science and Engineering** for their co-operation.

**ABSTRACT**

The main aim and objective of this project were to demonstrate the use of Asynchronous task in android asynchronous Task is designed to be a helper class around thread and handler and does not constitute a generic threading framework. Asynchronous Tasks are to ideally be 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 whose result is published on the UI thread, implement the use of asynchronous task by implementing the functionality of a simple moving banner using a set of start and stop buttons.

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **CHAPTERS Pg. No** | | | |
| **1.** |  | **INTRODUCTION** | **1-4** |
| **2.** |  | **PROBLEM STATEMENT AND OBJECTIVES** | **5** |
| **3.** |  | **SYSTEM REQUIREMENT** | **6** |
|  | 3.1 | Hardware Requirements | 6 |
|  | 3.2 | Software Requirements | 6 |
| **4.** |  | **STEPS FOR EXECUTION** | **7** |
| **5.** |  | **SOURCE CODES** | **8-11** |
|  | 5.1 | Activity\_main.xml | 8-9 |
|  | 5.2 | MainActivity.java | 9-11 |
| **6.** |  | **SNAPSHOTS** | **12-14** |
|  | 6.1 | Main Page | 12 |
|  | 6.2 | Start Operation | 13 |
|  | 6.3 | Stop Operation | 14 |
|  |  | **CONCLUSION** | **15** |
|  |  | **REFERENCES.** | **16** |

**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 added 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 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 some previous Android applications and access to large amounts of materials, we utilize the Java language, the Eclipse platform, Android ADT and the Android SDK to develop these three mobile applications. These systems have a nice interface and smooth operation. These Apps won’t steal any personal information but can exclude useless information and bring 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 touch screen mobile devices such as smartphones and tablets. Fig 1.1 represents android's user interface is based on direct manipulation, using touch gestures that loosely correspond to real-world actions, such as swiping, tapping and pinching, to manipulate on-screen objects, 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, each with a specialized 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.



Figure 1.1 Android image

Figure 1.1 shows the android logo, which is the robot consisting of a quadrangular torso.

## Software Development Kit:

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, you must download this software development kit. For example, if you want to create an Android app you require an SDK with java programming, for iOS apps you require an iOS SDK with swift language, and to develop MS Windows apps you require the.net language. 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 for the Android platform. Fig 1.2 represents Android image it 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 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 replaced Eclipse Android Development Tools (ADT) as Google's primary IDE for native Android application development.



Figure 1.3 Android Studio

Figure 1.3 shows the icon of android studio, help users identify your app and actions without it.

## Android Manifest:

The AndroidManifest.xml file *contains information of your package*, including components of the application such as activities, services, broadcast receivers, content providers etc.

It performs some other tasks also:

* + - It is responsible to protect the application to access any protected parts by providing the permissions.
    - It also declares the android api that the application is going to use.
    - It lists the instrumentation classes. The instrumentation classes provides profiling and other information. This information are removed just before the application is published etc.

## Main Activity:

The Main Activity File

The main activity code is a Java file MainActivity.java. This is the actual application file which ultimately gets converted to a Dalvik executable and runs your application. Following is the default code generated by the application wizard for Hello World! application −

package com.example.helloworld;

import android.support.v7.app.AppCompatActivity; import android.os.Bundle;

public class Main Activity extends App Compat Activity { @Override

protected void onCreate(Bundle saved Instance State) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main);

}}

Here, R.layout.activity\_main refers to the activity\_main.xml file located in the res/layout folder. The on Create() method is one of many methods that are figured when an activity is loaded.

**Chapter-2**

# PROBLEM STATEMENT AND OBJECTIVES

## 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 slide show.

## Objectives:

* + - To design an xml code that can display a moving banner of messages using the asynchronous task class.
    - To code a java program that can display a moving banner of messages using the asynchronous task class on press of a button.
    - Adding additional features to improve system and user interface by fading in and out of a changing background slide show.
    - Testing the code for all its intended functionalities by a android device or avd.

# Chapter-3

**SYSTEM REQUIREMENT**

## HARDWARE REQUIREMENT:

|  |  |  |
| --- | --- | --- |
| Minimum RAM | :- | 8 GB or more. |
| Processor | :- | Intel core i5 9th Gen. |
| Disk Space | :- | 20 GB of available disk space. |
| Display | :- | 1280 x 800 minimum screen resolution. |

* 1. **SOFTWARE REQUIREMENT:**

Operating System : - 64-bit Microsoft Windows 8/10. Emulator : - Pixel 4 API 30.

Target SDK version: - 30. Language Used : - xml, java.

JDK version : - java development kit 16.

**Chapter-4**

# STEPS FOR EXECUTION

## RUN ON EMULATOR:

* + 1. Run the Android Studio on laptop or desktop.
    2. Create a project and enter the source codes of xml and java.
    3. In Android Studio, create an Android Virtual Device (AVD) that the emulator can use to install and run your app.
    4. In the toolbar, select your app from the run/debug configurations drop-down menu.
    5. From the target device drop-down menu, select the AVD that you want to run your app on.
    6. Click Run . Android Studio installs the app on the AVD and starts the emulator.

**Chapter-5**

# SOURCE CODES

## XML CODE:

* 1. **Activity\_main.xml:**

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

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:padding="16dp"

tools:context=".MainActivity">

<TextView

android:id="@+id/bannerTextView"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Demonstration of Asynchronous Task"

android:textSize="24sp"

android:textStyle="bold" />

<Button

android:id="@+id/startButton"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@id/bannerTextView"

android:layout\_marginTop="16dp”

android:text="Start Task" />

<Button

android:id="@+id/stopButton"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@id/bannerTextView"

android:layout\_marginTop="16dp"

android:layout\_toEndOf="@id/startButton"

android:text="Stop Task" />

</RelativeLayout>

## JAVA CODE:

* 1. **MainActivity.java:**

import android.os.AsyncTask;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

private TextView bannerTextView;

private Button startButton;

private Button stopButton;

private BannerTask bannerTask;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

bannerTextView = findViewById(R.id.bannerTextView);

startButton = findViewById(R.id.startButton);

stopButton = findViewById(R.id.stopButton);

startButton.setOnClickListener(view -> startBannerTask());

stopButton.setOnClickListener(view -> stopBannerTask());

}

private void startBannerTask() {

if (bannerTask == null || bannerTask.isCancelled()) {

bannerTask = new BannerTask();

bannerTask.execute();

}

}

private void stopBannerTask() {

if (bannerTask != null) {

bannerTask.cancel(true);

bannerTask = null;

}

}

private class BannerTask extends AsyncTask<Void, Void, Void> {

private final String bannerText = "Demonstration of Asynchronous Task";

private final long bannerSpeed = 200L;

@Override

protected Void doInBackground(Void... params) {

while (!isCancelled()) {

publishProgress();

try {

Thread.sleep(bannerSpeed);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

return null;

}

@Override

protected void onProgressUpdate(Void... values) {

String currentText = bannerTextView.getText().toString();

String newText = currentText.substring(1) + currentText.charAt(0);

bannerTextView.setText(newText);

        }

     }

}

# CHAPTER-6

# SNAPSHOTS

## Main Page:

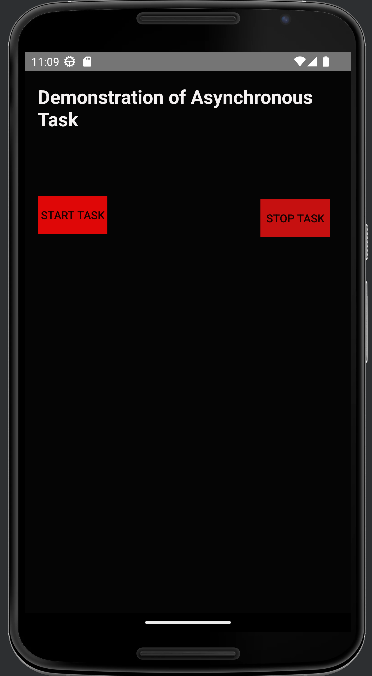


Figure 6.1 Main Page

Figure 6.1 represents the main page where it displays image icons and by clicking the buttons it performs operation, where it displays text and button.

## Start operation:

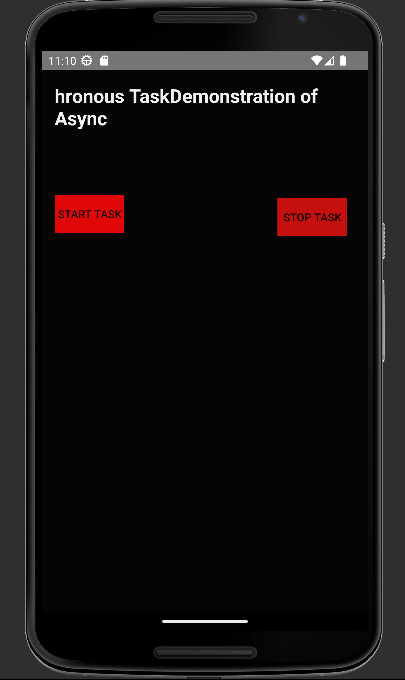


Figure 6.2 Home1 Page

Figure 6.2 shows on clicking on the start task button, the text starts to move right to left.

## 6.3 Stop operation:

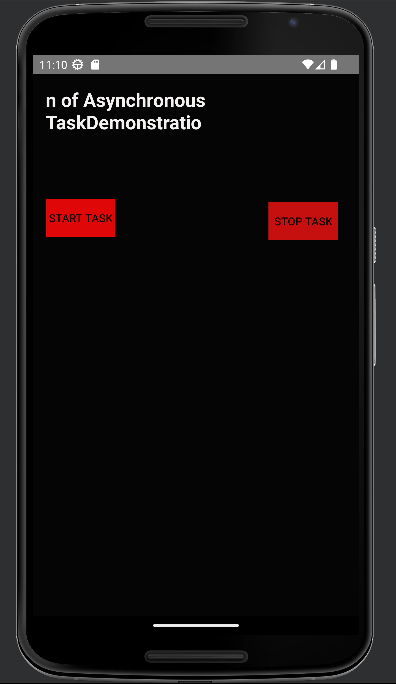


Figure 6.3 Home2 Page

Figure 6.3 shows on clicking stop task button, the text stops to move.

# CONCLUSION

Android as a full, open and free mobile device platform, with its powerful function and good user experience rapidly 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 experience of developing an android app is quite challenging, motivating as well as satisfying.

In the course of implementing this project we came across two highly usefull android studio classes and methods which are 1) asynchronous task which communicates with background threads and facilitates the background processing of task, and also to improvise the mini project we had to improvise on the background slide show which introduced us to the view flipper class which helps us in changing the background with fade\_in and fade\_out animations .

# REFERENCES

## Reference Book:

Google Developer Training, "Android Developer Fundamentals Course – Concept Reference”, Google Developer Training Team, 2017.

## Website:

[1]. YouTube: <https://www.youtube.com/> [2]. h[ttps://developer.android.com/studio](https://developer.android.com/studio)

[3]. <https://github.com/>

[4].https:/[/www.g](http://www.gitbook.com/book/google-developer-training/android-developer-)i[tbook.com/book/google-developer-training/android-developer-](http://www.gitbook.com/book/google-developer-training/android-developer-) [5].fundamentals-course- concepts/details [6].https://developer.android.com/reference/android/os/AsyncTask