

## 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 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 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.

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 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 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.

## 1.1 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, 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.



Fig 1.1 Android image

## 1.2 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 have to 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

## 1.3 Android Studio:

Android Studio is an integrated development environment (IDE) for developing for the Android platform. 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..



Fig 1.3 Android Studio

### 1.4 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 provide profiling and other information. This information is removed just before the application is published.

### 1.5 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 AppCompatActivity {
@Override protected void onCreate(Bundle
savedInstanceState) {
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 onCreate() method is one of many methods that are figured when an activity is loaded

## 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 slide show.

#### 2.2 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****3.1 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.

**3.2 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****4.1 RUN ON EMULATOR:**

- Run the Android Studio on laptop or desktop.
- Create a project and enter the source codes of xml and java.

- In Android Studio, create an Android Virtual Device (AVD) that the emulator can use to install and run your app.
- In the toolbar, select your app from the run/debug configurations drop-down menu.
- From the target device drop-down menu, select the AVD that you want to run your app on.
- Click Run . Android Studio installs the app on the AVD and starts the emulator

## Chapter-5

## SOURCE CODES

### XML CODE:

#### 5.1 Activitymain.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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"
    tools:context="com.example.SmsApp.MainActivity"
    android:background="@drawable/wallpaper">

    <TextView
        android:id="@+id/txtnumber"
        android:layout_width="229dp"
        android:layout_height="29dp"
        android:text="Phone Number"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.136"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/txtmessage"
        android:layout_width="327dp"
        android:layout_height="33dp"
        android:text="Message Content"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.295"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
```

```
        app:layout_constraintVertical_bias="0.545" />

        <TextView
            android:id="@+id/txtTimestamp"
            android:layout_width="136dp"
            android:layout_height="19dp"
            android:text="Timestamp"
            android:textColor="@android:color/black"
            android:textSize="14sp"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintHorizontal_bias="0.948"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent"
            app:layout_constraintVertical_bias="0.445"
            tools:ignore="MissingConstraints" />

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

## 5.2 AndroidManifest.xml:

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

    <uses-permission android:name="android.permission.RECEIVE_SMS"></uses-
permission>
    <uses-permission android:name="android.permission.READ_SMS" ></uses-
permission>
    <uses-permission android:name="android.permission.SEND_SMS"></uses-
permission>
    <uses-permission android:name="android.permission.RECEIVE_SMS" />
    j1

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher_foreground"
        android:label="SmsApp"
        android:roundIcon="@mipmap/ic_launcher_foreground"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name="com.example.SmsApp.MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER"
/>
            </intent-filter>
        </activity>
        <receiver android:name="com.example.SmsApp.MySMSReceiver"
            android:enabled="true"
            android:exported="true"
            android:permission="android.permission.BROADCAST_SMS">
            <intent-filter android:priority="999">
                <action
android:name="android.provider.Telephony.SMS_RECEIVED" />
            </intent-filter>
        </receiver>
```



```
</application>

</manifest>
```

## 5.3 Main Activity:

```
package com.example.SmsApp;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.Manifest;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.widget.TextView;
import android.widget.Toast;

import com.example.SmsApp.R;

import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Locale;

public class MainActivity extends AppCompatActivity {

    private static final int MY_PERMISSION_REQUEST_RECEIVE_SMS = 0;
    TextView txtnumber, txtmessage, txtTimestamp;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        if (ContextCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.RECEIVE_SMS) != PackageManager.PERMISSION_GRANTED) {
            if
(ActivityCompat.shouldShowRequestPermissionRationale(MainActivity.this,
Manifest.permission.RECEIVE_SMS)) {
                // Do not do anything
            } else {
                ActivityCompat.requestPermissions(MainActivity.this, new
String[]{Manifest.permission.RECEIVE_SMS},
MY_PERMISSION_REQUEST_RECEIVE_SMS);
            }
        }

        Bundle b = getIntent().getBundleExtra("data");
        txtnumber = findViewById(R.id.txtnumber);
        txtmessage = findViewById(R.id.txtmessage);
        txtTimestamp = findViewById(R.id.txtTimestamp);

        if (b != null) {

            String number = b.getString("number");
            String content = b.getString("content");
```

```
        txtnumber.setText(number);
        txtmessage.setText(content);

        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss", Locale.getDefault());

        // Get the current system date and time
        String currentDateTimeString = dateFormat.format(new Date());

        txtTimestamp.setText(currentDateTimeString);
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, String
permissions[], int[] grantResults) {
        switch (requestCode) {
            case MY_PERMISSION_REQUEST_RECEIVE_SMS: {
                if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                    Toast.makeText(this, "Thank you for permitting",
Toast.LENGTH_LONG).show();
                } else {
                    Toast.makeText(this, "Can't do anything until you
permit me", Toast.LENGTH_LONG).show();
                }
            }
        }
    }
}
```

## 5.4 SMSReceiver:

```
package com.example.SmsApp;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.telephony.SmsMessage;
import android.widget.Toast;

public class MySMSReceiver extends BroadcastReceiver {

    private static final String SMS_RECEIVED =
"android.provider.Telephony.SMS_RECEIVED";
    private static final String TAG = "SmsBroadcastReceiver";

    @Override
    public void onReceive(Context context, Intent intent) {

        if(intent.getAction() == SMS_RECEIVED)
        {
            Bundle bundle = intent.getExtras();
            SmsMessage[] smsg = null;
            String sms_str = "";
            if(bundle != null)
```

```
{
    Object[] pdus = (Object[]) bundle.get("pdus");
    smsm = new SmsMessage[pdus.length];
    String senderNumber = null;
    String senderMessage = null;
    for(int i = 0; i < pdus.length; i++) {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            String format = bundle.getString("format");
            smsm[i] = SmsMessage.createFromPdu((byte[])
pdus[i], format);
        } else {
            smsm[i] = SmsMessage.createFromPdu((byte[])
pdus[i]);
        }

        Bundle b1 = new Bundle();
        b1.putString("number",
smsm[i].getOriginatingAddress());
        b1.putString("content", smsm[i].getMessageBody());

        Intent smsIntent = new Intent(context,
MainActivity.class);
        smsIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        smsIntent.putExtra("data", b1);
        context.startActivity(smsIntent);
    }
    Toast.makeText(context,sms_str ,Toast.LENGTH_LONG).show();
}
}
```

## Chapter-6

## SNAPSHOTS

### 6.1 Permission requesting :

This is the permission requesting page of the SMS application. Here user can allow or deny permission for the sms application. For further usage of sms application

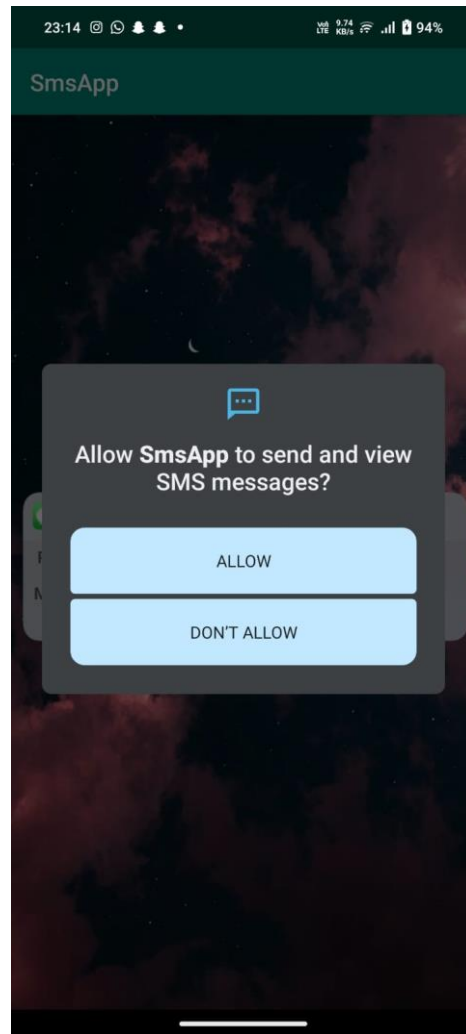


Fig 6.1 Android Studio

## 6.2 Main page:

This is the main application page where user can receive the messages

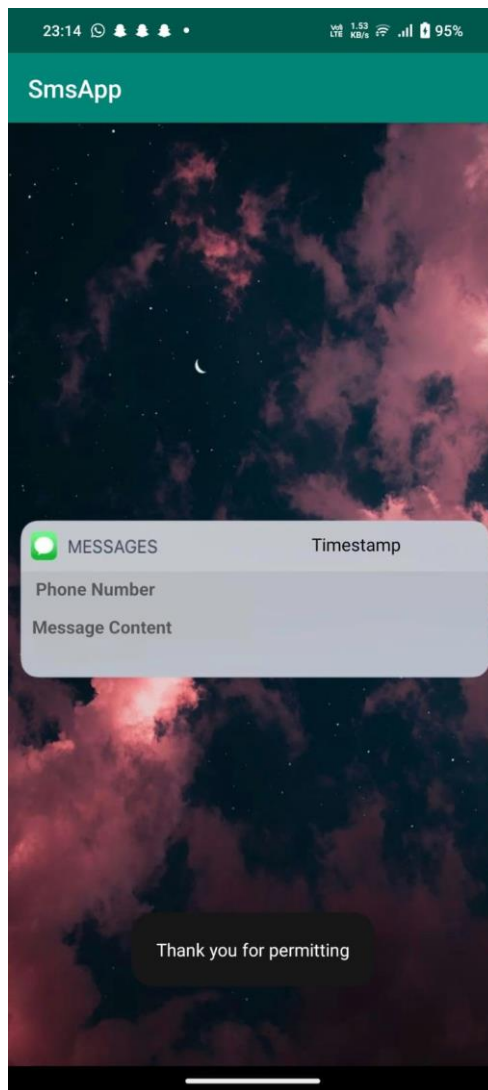


Fig 6.2 Android Studio

### 6.3 SMS received successfully :

Here we can see message successfully recieved as toast message, which means the SMS is recieved to the user phone number. In this page user can view senders message and phone number.

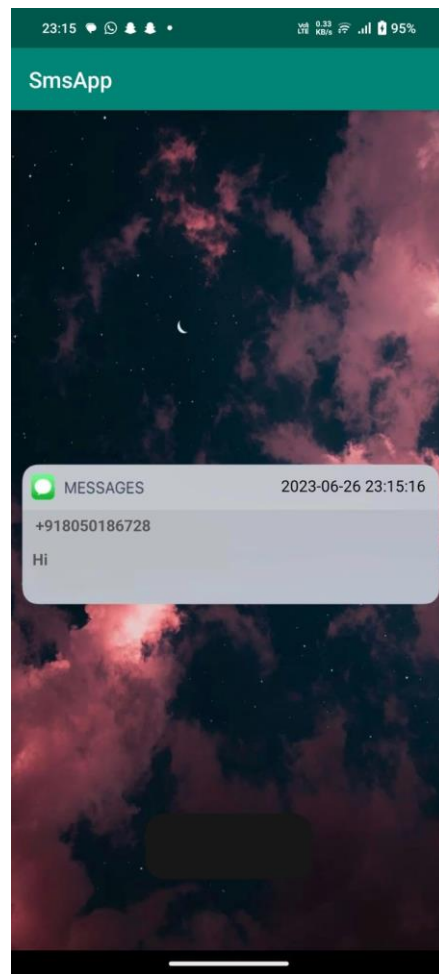


Fig 6.3 Android Studio

## CONCLUSION AND FUTURE ENHANCEMENT

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.

## REFERENCES

### Reference Book :

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

### Website:

[1]. YouTube : <https://www.youtube.com/>

\*Coding With Evan

[2]. <https://developer.android.com/studio>

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

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

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

[6]. <https://developer.android.com/reference/android/os/Messagingappliance>

