



Dr. Babasaheb Ambedkar Open University

(Established by Government of Gujarat)

PGDMAD-203

Software Lab For Advance Android Mobile Application



Advanced Android Mobile Application

Post Graduate Diploma in Mobile Application Development

2019

Software Lab for Advanced Android Mobile Application

Dr. Babasaheb Ambedkar Open University



Expert Committee

Prof. (Dr.) Nilesh Modi Professor and Director, School of Computer Science, Dr. Babasaheb Ambedkar Open University, Ahmedabad	(Chairman)
Prof. (Dr.) Ajay Parikh Professor and Head, Department of Computer Science Gujarat Vidyapith, Ahmedabad	(Member)
Prof. (Dr.) Satyen Parikh Dean, School of Computer Science and Application Ganpat University, Kherva, Mahesana	(Member)
Prof. M. T. Savaliya Associate Professor and Head, Computer Eng. Department Vishwakarma Engineering College, Ahmedabad	(Member)
Dr. Himanshu Patel Assistant Professor, School of Computer Science, Dr. Babasaheb Ambedkar Open University, Ahmedabad	(Member Secretary)

Course Writer

Dr. Chetan Gondaliya Assistant Professor, Department of Computer Science, Ganpat University, Kherva
Dr. Hiral Patel, Assistant Professor, Department of Computer Science, Ganpat University, Kherva
Dr. Ashishkumar Parejiya Assistant Professor, Institute of Information and Communication Technology, Indus University, Ahmedabad

Subject Reviewer

Dr. Himanshu Patel Assistant Professor, School of Computer Science, Dr. Babasaheb Ambedkar Open University, Ahmedabad
--

Editors

Prof. (Dr.) Nilesh Modi Professor and Director, School of Computer Science, Dr. Babasaheb Ambedkar Open University, Ahmedabad
Dr. Himanshu Patel Assistant Professor, School of Computer Science, Dr. Babasaheb Ambedkar Open University, Ahmedabad

Acknowledgement: “The content in this book is modifications based on work created and shared by the Android Open Source Project and used according to terms described in the Creative Commons 2.5 Attribution License.”



June 2019, © Dr. Babasaheb Ambedkar Open University

ISBN-978-81-940577-5-8

Printed and published by: Dr. Babasaheb Ambedkar Open University, Ahmedabad



Software Lab for Advanced Android Mobile Application Development

List of Programs

Practical-1: Content Provider	01
Practical-2: Shared Preference	04
Practical-3: JSON and Web Service	07
Practical-4: ImageView	12
Practical-5: Image Gallery	16
Practical-6: Drawing 2D	22
Practical-7: 3D Graphics	24
Practical-8: Audio Player	38
Practical-9: Animation	47
Practical-10: Downloading a file	71
Practical-11: WebView	79
Practical-12: Wi-Fi Connectivity	84
Practical-13: Telephony API	88
Practical-14 Search View	92
Practical-15 Location Mapping	95
Practical-16: Bluetooth	105
Practical-17: Connecting with Facebook	122
Practical -18 Sensors	130
Practical-19: NFC	138
Practical-20: Speech and Gestures Recognizer	141

Content Providers

1

MainActivity.java

```
package com.example.d.contentprovider;

import android.content.ContentResolver;
import android.database.Cursor;
import android.provider.ContactsContract;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.ListView;
import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {
    ListView lst;
    ArrayList<String> aa = new ArrayList<String>();

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

        lst=findViewById(R.id.lst);

        findViewById(R.id.btn).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                ContentResolver contentResolver = getContentResolver();
                Cursor c =
                contentResolver.query(ContactsContract.Contacts.CONTENT_URI,
                    new String[]{ContactsContract.Contacts.DISPLAY_NAME_PRIMARY,
                ContactsContract.Contacts._ID,ContactsContract.Contacts.HAS_PHONE_NUMBER
                },
                    null, null, null);
                if (c != null && c.getCount() > 0) {

                    for(int i=0;i<c.getCount();i++) {

                        c.moveToNext();
                        if (c.getInt(2) == 1) {
```



```

    Cursor phone =
    contentResolver.query(ContactsContract.CommonDataKinds.Phone.CONTENT_URI
    ,
        null,
    ContactsContract.CommonDataKinds.Phone.CONTACT_ID + " = " + c.getInt(1), null,
    null);

        if(phone != null && phone.getCount() >0) {
            phone.moveToNext();
            aa.add(c.getString(0) + "=" +
phone.getString(phone.getColumnIndex(ContactsContract.CommonDataKinds.Phon
e.NUMBER)));
            phone.close();
        }else{
            aa.add(c.getString(0));
        }
        } else {
            aa.add(c.getString(0));
        }
    }
}

    ArrayAdapter<String> ad = new ArrayAdapter<String>(MainActivity.this,
R.layout.support_simple_spinner_dropdown_item, aa);
    lst.setAdapter(ad);
}
});
}
}
}

```

activity_main.xml :

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:orientation="vertical"
    tools:context="com.example.d.contentprovider.MainActivity">
    <Button
        android:id="@+id/btn"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Load Contacts"
        android:textStyle="bold" />

    <ListView
        android:id="@+id/lst"

```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
    </ListView>
```

```
</LinearLayout>
```

AndroidManifest:

```
<?xml version="1.0" encoding="utf-8" ?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.d.contentprovider">
    <uses-permission android:name="android.permission.READ_CONTACTS" />
    <uses-permission android:name="android.permission.WRITE_CONTACTS" />
    <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/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Output:



Figure-3 Output of Program

Shared Preferences

MainActivity:

```
package com.example.d.sharedpreference;

import android.content.SharedPreferences;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

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

        SharedPreferences pref =
        getApplicationContext().getSharedPreferences("MyPref", MODE_PRIVATE);
        SharedPreferences.Editor editor = pref.edit();

        String s=pref.getString("email", null);

        if(s==null) {

            editor.putString("email", "xyz@gmail.com");
            editor.commit();
            Toast.makeText(this, "Sucesfully Registered",
            Toast.LENGTH_LONG).show();

        }else{
            Toast.makeText(this, "Already Registered", Toast.LENGTH_SHORT).show();
        }
    }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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.d.sharedpreference.MainActivity">
```

```
<EditText
    android:id="@+id/txt1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_above="@+id/txt2"
    android:layout_alignParentStart="true"
    android:layout_marginBottom="48dp"
    android:hint="emailID"
    android:textSize="20dp" />
```

```
<EditText
    android:id="@+id/txt2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_above="@+id/btn"
    android:layout_alignParentStart="true"
    android:layout_marginBottom="93dp"
    android:hint="Password"
    android:textSize="20dp" />
```

```
<Button
    android:id="@+id/btn"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentStart="true"
    android:layout_marginBottom="154dp"
    android:text="Log in" />
```

```
</RelativeLayout>
```

MainActivity2.java

```
package com.example.d.sharedpreference;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class Main2Activity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
    }
}
```

activity_main2.xml

```
package com.example.d.sharedpreference;
```

```
import android.support.v7.app.AppCompatActivity;
```

```
import android.os.Bundle;
```

```
public class Main2Activity extends AppCompatActivity {
```

```
    @Override
```

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

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main2);
```

```
    }
```

```
}
```

JSON Web Service

3

Step-1 Create new project.

Open build.gradle and add the following dependency:

```
dependencies
{
    implementation 'com.android.volley:volley:1.0.0'
}
```

Step-2 In AndroidManifest.xml add the internet permission:

```
<uses-permission
```

```
    android:name="android.permission.INTERNET />"
```

Perform such code in MainActivity.java

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonArrayRequest;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.StringRequest;
import com.example.devatsi.jsonnew.app.AppController;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
```

```
public class MainActivity extends AppCompatActivity {
```

```
    Button btn1, btn2, btn3;
    TextView txt1;
```

```
    public static String URL_LOCAL_JSON_OBJECT =
"http://192.168.1.101/webservice_jsonobject.php";
    public static String URL_LOCAL_JSON_ARRAY =
"http://192.168.1.101/webservice_jsonarray.php";
    private String jsonResponse;
```

```

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

    btn1 = (Button) findViewById(R.id.btn1);
    btn2 = (Button) findViewById(R.id.btn2);
    btn3 = (Button) findViewById(R.id.btn3);

    txt1 = (TextView) findViewById(R.id.txt);

    btn1.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            makejsonarrayrequest();
        }
    });

    btn2.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            makejsonobjectrequest();
        }
    });

    btn3.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            makejsonstringobject();
        }
    });
}

void makejsonarrayrequest() {
    final JSONArrayRequest jsonArrayRequest = new
    JSONArrayRequest(URL_LOCAL_JSON_ARRAY,
        new Response.Listener<JSONArray>() {
            @Override
            public void onResponse(JSONArray response) {

                Toast.makeText(MainActivity.this, response.toString(),
                Toast.LENGTH_SHORT).show();

                try {
                    jsonResponse = "";
                    for (int i = 0; i < response.length(); i++) {

                        JSONObject student = (JSONObject) response.get(i);

```

```

        String sno = student.getString("sno");
        String name = student.getString("sname");
        String course = student.getString("city");

        jsonResponse += "Sno: " + sno + "\n\n";
        jsonResponse += "Sname: " + name + "\n\n";
        jsonResponse += "City: " + course + "\n\n";

        txt1.setText(jsonResponse);
    }

    } catch (JSONException e) {
        e.printStackTrace();
        Toast.makeText(getApplicationContext(),
            "Error: " + e.getMessage(),
            Toast.LENGTH_LONG).show();
    }
}
}, new Response.ErrorListener() {
@Override
public void onErrorResponse(VolleyError error) {
    Toast.makeText(MainActivity.this, error.getMessage(),
        Toast.LENGTH_SHORT).show();
}
});

AppController.getInstance().addToRequestQueue(jsonArrayRequest);
}

void makejsonobjectrequest() {
    JsonObjectRequest jsonObjectRequest = new
    JsonObjectRequest(Request.Method.GET,
        URL_LOCAL_JSON_OBJECT, null, new
    Response.Listener<JSONObject>() {
        @Override
        public void onResponse(JSONObject response) {

            try {
                String name = response.getString("sno");
                String email = response.getString("sname");
                String home = response.getString("city");

                jsonResponse = "";
                jsonResponse += "Sno: " + name + "\n\n";
                jsonResponse += "Sname: " + email + "\n\n";
                jsonResponse += "City: " + home + "\n\n";

                txt1.setText(jsonResponse);
            }
        }
    });
}

```



```

        } catch (JSONException e) {
            e.printStackTrace();
            Toast.makeText(getApplicationContext(),
                "Error: " + e.getMessage(),
                Toast.LENGTH_LONG).show();
        }
    }
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {

        Toast.makeText(getApplicationContext(),
            error.getMessage(), Toast.LENGTH_SHORT).show();
    }
});

AppController.getInstance().addToRequestQueue(jsonObjectRequest);
}

void makejsonstringobject(){
    StringRequest stringRequest = new StringRequest(Request.Method.POST,
URL_LOCAL_JSON_ARRAY,
        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                Log.d("Response", response);

                txt1.setText(response);

            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                Toast.makeText(getApplicationContext(),
                    error.getMessage(), Toast.LENGTH_SHORT).show();
            }
        });
AppController.getInstance().addToRequestQueue(stringRequest);
}
}

```

Appcontroller.java

```
import android.app.Application;
import android.text.TextUtils;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.toolbox.Volley;

public class AppController extends Application {

    public static final String TAG = AppController.class.getSimpleName();
    private RequestQueue mRequestQueue;
    private static AppController mInstance;

    @Override
    public void onCreate() {
        super.onCreate();
        mInstance = this;
    }

    public static synchronized AppController getInstance() {
        return mInstance;
    }

    public RequestQueue getRequestQueue() {
        if (mRequestQueue == null) {
            mRequestQueue = Volley.newRequestQueue(getApplicationContext());
        }

        return mRequestQueue;
    }

    public <T> void addToRequestQueue(Request<T> req, String tag) {
        req.setTag(TextUtils.isEmpty(tag) ? TAG : tag);
        getRequestQueue().add(req);
    }

    public <T> void addToRequestQueue(Request<T> req) {
        req.setTag(TAG);
        getRequestQueue().add(req);
    }

    public void cancelPendingRequests(Object tag) {
        if (mRequestQueue != null) {
            mRequestQueue.cancelAll(tag);
        }
    }
}
```

ImageView

4

Step 1: Create a new project and name it ImageViewExample. In this step we create a new project in android studio by filling all the necessary details of the app like app name, package name, api versions etc.

Select File -> New -> New Project and Fill the forms and click "Finish" button.

Step 2: Download any two images lion and monkey from the web. Now save those images in the drawable folder of your project.

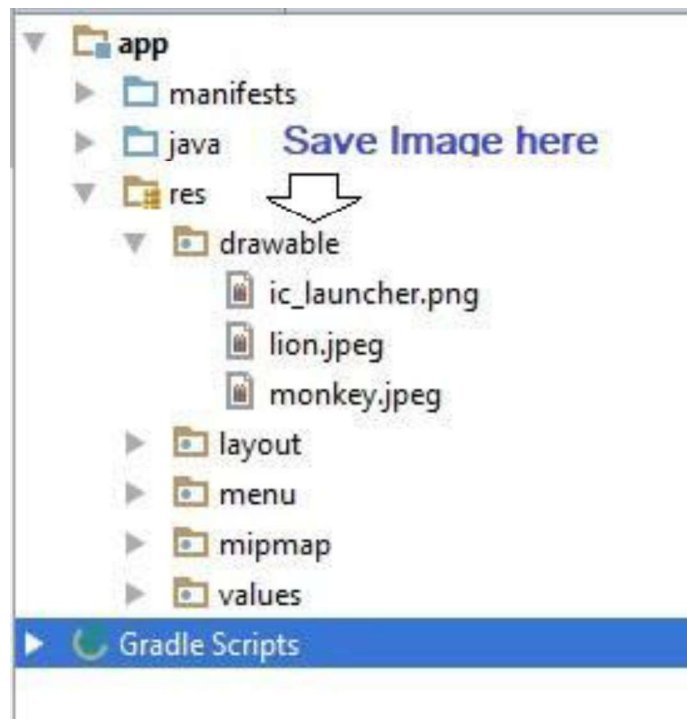


Figure-12 Upload or save images here

Step 3: Now open res -> layout -> activity_main.xml (or) main.xml and add following code:

In this step we add the code for displaying an image view on the screen in a relative layout. Here make sure you have already saved two images name lion and monkey in your drawable folder.

```
<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:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
tools:context=".MainActivity">
```

```
<ImageView
    android:id="@+id/simpleImageViewLion"
    android:layout_width="fill_parent"
    android:layout_height="200dp"
    android:scaleType="fitXY"
    android:src="@drawable/lion" />
```

```
<ImageView
    android:id="@+id/simpleImageViewMonkey"
    android:layout_width="fill_parent"
    android:layout_height="200dp"
    android:layout_below="@+id/simpleImageViewLion"
    android:layout_marginTop="10dp"
    android:scaleType="fitXY"
    android:src="@drawable/monkey" />
```

```
</RelativeLayout>
```

Step 4: Now open app -> java -> package -> MainActivity.java and add the following code:

In this step we add the code to initiate the image view's and then perform click event on them.

```
package example.baou.imageviewexample;
import android.graphics.Color;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
```

```

import android.view.MenuItem;
import android.view.View;
import android.widget.ImageView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

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

        ImageView simpleImageViewLion = (ImageView)
            findViewById(R.id.simpleImageViewLion);

        //get the id of first image view

        ImageView simpleImageViewMonkey = (ImageView)
            findViewById(R.id.simpleImageViewMonkey);

        //get the id of second image view

        simpleImageViewLion.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Toast.makeText(getApplicationContext(), "Lion",
                    Toast.LENGTH_LONG).show();

                //display the text on image click event
            }
        });

        simpleImageViewMonkey.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Toast.makeText(getApplicationContext(), "Monkey",
                    Toast.LENGTH_LONG).show();

                //display the text on image click event
            }
        });
    }
}

```

Now start AVD in Emulator and run the App. You will see the images of Lion and Monkey displayed on screen. Click on any Animal image and his name will appear on Screen. We clicked on Lion.



Output of Imagesview code (Clicked on Lion, its show's toast message as lion)

Creating the Image Gallery App

We will be using 2 popular libraries in our gallery.

Glide – powerful, yet simple to use image loading library by Square.Inc

PhotoView – flexible ImageView that supports gestures such as zoom and pan

So first, let's start by adding the above dependencies. Add the following lines to your app's build.gradle file.

```
repositories {
    maven {
        url "https://baou.edu.in"
    }
}
dependencies {
    ...
    compile 'com.squareup.picasso:picasso:2.5.2'
    compile 'com.github.chrisbanes:PhotoView:2.0.0'
}
```

First off, we need to create a model class for our images.

```
public class ImageModel implements Parcelable {
    private String name, url;
    //TODO: Add your Getters and setters here.
}
```

Next up, we need to create our Activity. The only one that we will be using in this chapter. We will use Fragments to handle everything else. We will create an 'Empty Activity' using Android Studio. Let's call it MainActivity as per convention.

Utility class for Image loading Before we go ahead and code, we'll create a utility class. This will provide us with the image data. We could keep this code inside an Activity too, but using a separate utility class helps keep our code clean.

```
public class Utils {
    static String IMGs[] = {
        "https://images.unsplash.com/photo-1444090542259-0af8fa96557e",
        "https://images.unsplash.com/photo-1439546743462-802cabef8e97",
        "https://images.unsplash.com/photo-1441155472722-d17942a2b76a",
        //more image links
    }
}
```

```

    };
    public static ArrayList<ImageModel> getData() {
        ArrayList<ImageModel> arrayList = new ArrayList<>();
        for (int i = 0; i < IMGs.length; i++) {
            ImageModel imageModel = new ImageModel();
            imageModel.setName("Image " + i);
            imageModel.setUrl(IMGs[i]);
            arrayList.add(imageModel);
        }
        return arrayList;
    }
}
}
}

```

Defining Click Interface for Gallery Images

To handle clicks on images in the gallery, we will create a simple interface. This will help listen to click events.

```

    public interface GalleryItemClickListener {
        void onGalleryItemClickListener(int position, ImageModel imageModel,
            ImageView imageView);
    }

```

Creating the Image Gallery Adapter

Next, we need to create an adapter that will handle the RecyclerView data and display it.

RecyclerView offers powerful list and grid display options via LayoutManager.

So, here's how we define our RecyclerView.Adapter. It's very basic. Almost like how every standard adapter would be.

```

public class GalleryAdapter extends
RecyclerView.Adapter<GalleryAdapter.GalleryViewHolder> {

    private final GalleryItemClickListener galleryItemClickListener;
    private ArrayList<ImageModel> galleryList;

    public GalleryAdapter(ArrayList<ImageModel> galleryList,
        GalleryItemClickListener galleryItemClickListener) {
        this.galleryList = galleryList;
        this.galleryItemClickListener = galleryItemClickListener;
    }

    @Override
    public GalleryViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {
        return new GalleryViewHolder(LayoutInflater.from(parent.getContext())
.inflate(R.layout.gallery_item, parent, false));
    }
}

```



```

    }

    @Override
    public void onBindViewHolder(final GalleryViewHolder holder, int position) {
        final ImageModel imageModel = galleryList.get(position);

        Glide.with(holder.galleryImageView.getContext()).
            load(imageModel.getUrl())
            .thumbnail(0.5f)
            .diskCacheStrategy(DiskCacheStrategy.ALL)
            .into(holder.galleryImageView);

        holder.galleryImageView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                galleryItemClickListener.onGalleryItemClickListener(holder.getAdapterPosition(),
                    imageModel, holder.galleryImageView);
            }
        });
    }

    @Override
    public int getItemCount() {
        return galleryList.size();
    }

    static class GalleryViewHolder extends RecyclerView.ViewHolder {
        private ImageView galleryImageView;

        GalleryViewHolder(View view) {
            super(view);
            galleryImageView = (ImageView) view.findViewById(R.id.galleryImage);
        }
    }
}

```

The layout of the RecyclerView items is as follows.

```

<ImageView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/galleryImage"
    android:layout_width="match_parent"
    android:layout_height="128dp"
    android:layout_margin="1dp"
    android:adjustViewBounds="true"
    android:scaleType="centerCrop" />

```

Nothing fancy till now. We have created a regular RecyclerView.Adapter. This uses an interface to handle click events. Then we use Glide to load the image into our ImageView.

You can however save yourself trouble and NOT write RecyclerView.Adapter boilerplate code.

Using Fragments instead of Activities

Now that our Adapter is complete, we need to create a Fragment to hold our RecyclerView.

```
public class RecyclerViewFragment extends Fragment {
    public static final String TAG = RecyclerViewFragment.class.getSimpleName();
    public RecyclerViewFragment() {
    }
    public static RecyclerViewFragment newInstance() {
        return new RecyclerViewFragment();
    }
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_recycler_view, container, false);
    }
    @Override
    public void onViewCreated(View view, @Nullable Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);
        GalleryAdapter galleryAdapter = new GalleryAdapter(Utils.getData(),this);
        RecyclerView recyclerView = (RecyclerView)
view.findViewById(R.id.recycler_view);
        GridLayoutManager gridLayoutManager = new
GridLayoutManager(getContext(), 2);
        recyclerView.setLayoutManager(gridLayoutManager);
        recyclerView.setAdapter(galleryAdapter);
    }
}
```

In the onViewCreated() method, we create our RecyclerView.Adapter (GalleryAdapter). Notice that Utils.getData() provides a list of images which we populated.

Next, we create a `GridLayoutManager` which consists of two columns. Finally, we attach the adapter and `LayoutManager` to `RecyclerView`.

Another type of `GridLayoutManger` is the `StaggeredGridLayoutManager`. Here's the Fragment XML layout.

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <android.support.v7.widget.RecyclerView
        android:id="@+id/recycler_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:clipToPadding="false" />

</LinearLayout>
```

Now, let's launch the `RecyclerViewFragment` from our `MainActivity`.

```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        getSupportFragmentManager()
            .beginTransaction()
            .add(R.id.content, RecyclerViewFragment.newInstance())
            .commit();
    }
}
```

Our gallery view is ready. Now if you run the code, you should get something like this.

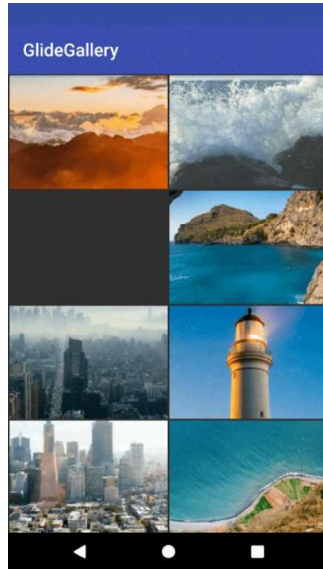


Figure -16 Output of gallery company

you've built a complete image gallery app using above code and along the way you've learned how the library works and how you can integrate it into your own project.

You've also learned how to display both local and remote images, tagging requests, prioritizing requests, and how to apply image transformations like resizing. Not only that, but you've seen how easy it is to enable and disable caching, error handling, and custom request listeners.

Example

Create a new Java class that should extend from View class. Override the onDraw() method. In this method, you can use Canvas class to draw the different shapes.

File name: MyView.java

```
public class MyView extends View
{
    public MyView(Context context)
    {
        super(context);
        // TODO Auto-generated constructor stub
    }
    @Override
    protected void onDraw(Canvas canvas)
    {
        // TODO Auto-generated method stub
        super.onDraw(canvas);
        int radius;
        radius = 50;
        Paint paint = new Paint();
        paint.setStyle(Paint.Style.FILL);
        paint.setColor(Color.parseColor("#CD5C5C"));
        canvas.drawCircle(150,200, radius, paint);
        canvas.drawRoundRect(new RectF(20,20,100,100), 20, 20, paint);
        canvas.rotate(-45);
        canvas.drawText("BAOU", 40, 180, paint);
        canvas.restore();
    }
}
```

File name: MainActivity.java

Note: You have to pass the object of subclass that extends from View class in setContentView() method as given below. In our case the name of the subclass is MyView.

```
Public class MainActivity extends Activity
{
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(new MyView(this));
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu)
    {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
```

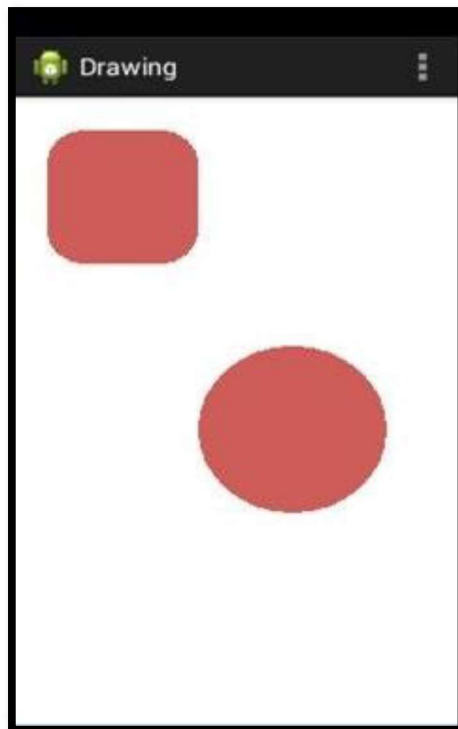


Figure-17 2D Object Output

3D Graphics

7

1. Create a Project Graphics3d.
2. Put an image in res/drawable.
3. Create a custom view MyGLView.java ,which will look like this:

Create Android project and setup accordingly apps configuration:

```
package com.app.Graphics3d;
import android.content.Context;
import android.opengl.GLSurfaceView;
class MyGLView extends GLSurfaceView {
    private final MyGLRenderrer renderrer;
    MyGLView(Context context) {
        super(context);
        renderrer = new MyGLRenderrer(context);
        setRenderrer(renderrer);
    }
}
```

```
Package com.app.Graphics3d;
import android.content.Context;
import android.opengl.GLSurfaceView;
class MyGLView extends GLSurfaceView {
    private final MyGLRenderrer renderrer;
    MyGLView(Context context) {
        super(context);
        renderrer = new MyGLRenderrer(context);
        setRenderrer(renderrer);
    }
}
```

4. In OpenGL ES on Android, drawing is separated out into a rendering class which is responsible for initializing and drawing the entire screen.
5. Create MyGLRenderrer.java ,which will look like this:

```
package com.app.Graphics3d;
```

```

import javax.microedition.khronos.egl.EGLConfig;
import javax.microedition.khronos.opengles.GL10;
import android.content.Context;
import android.opengl.GLSurfaceView;
import android.opengl.GLU;
import android.util.Log;

class MyGLRenderer implements GLSurfaceView.Renderer {
    private static final String TAG = "GLRenderer";
    private final Context context;
    private final MyCube cube = new MyCube();
    private long startTime;
    private long fpsStartTime;
    private long numFrames;
    MyGLRenderer(Context context) {
        this.context = context;
    }

    public void onSurfaceCreated(GL10 gl, EGLConfig config)
    {
        boolean SEE_THRU = true;

        startTime = System.currentTimeMillis();
        fpsStartTime = startTime;
        numFrames = 0;
        // Define the lighting
        float lightAmbient[] = new float[] { 0.2f, 0.2f, 0.2f, 1 };
        float lightDiffuse[] = new float[] { 1, 1, 1, 1 };
        float[] lightPos = new float[] { 1, 1, 1, 1 };
        gl.glEnable(GL10.GL_LIGHTING);
        gl.glEnable(GL10.GL_LIGHT0);
        gl.glLightfv(GL10.GL_LIGHT0, GL10.GL_AMBIENT, lightAmbient, 0);
        gl.glLightfv(GL10.GL_LIGHT0, GL10.GL_DIFFUSE, lightDiffuse, 0);
        gl.glLightfv(GL10.GL_LIGHT0, GL10.GL_POSITION, lightPos, 0);
        // What is the cube made of?
        float matAmbient[] = new float[] { 1, 1, 1, 1 };

```



```

        float matDiffuse[] = new float[] { 1, 1, 1, 1 };
        gl.glMaterialfv(GL10.GL_FRONT_AND_BACK, GL10.GL_AMBIENT,
            matAmbient, 0);
        gl.glMaterialfv(GL10.GL_FRONT_AND_BACK, GL10.GL_DIFFUSE,
            matDiffuse, 0);
        // Set up any OpenGL options we need
        gl.glEnable(GL10.GL_DEPTH_TEST);
        gl.glDepthFunc(GL10.GL_LEQUAL);
        gl.glEnableClientState(GL10.GL_VERTEX_ARRAY);

        if (SEE_THRU) {
            gl.glDisable(GL10.GL_DEPTH_TEST);
            gl.glEnable(GL10.GL_BLEND);
            gl.glBlendFunc(GL10.GL_SRC_ALPHA, GL10.GL_ONE);
        }

// Enable textures
        gl.glEnableClientState(GL10.GL_TEXTURE_COORD_ARRAY);
        gl.glEnable(GL10.GL_TEXTURE_2D);

        // Load the cube's texture from a bitmap
        MyCube.loadTexture(gl, context, R.drawable.one);
    }

    public void onSurfaceChanged(GL10 gl, int width, int height)
    {
        // Define the view frustum
        gl.glViewport(0, 0, width, height);
        gl.glMatrixMode(GL10.GL_PROJECTION);
        gl.glLoadIdentity();
        float ratio = (float) width / height;
        GLU.gluPerspective(gl, 45.0f, ratio, 1, 100f);
    }

    public void onDrawFrame(GL10 gl) {
        // Clear the screen to black
        gl.glClear(GL10.GL_COLOR_BUFFER_BIT

```

```

        | GL10.GL_DEPTH_BUFFER_BIT);

        // Position model so we can see it
        gl.glMatrixMode(GL10.GL_MODELVIEW);
        gl.glLoadIdentity();
        gl.glTranslatef(0, 0, -3.0f);

        // Set rotation angle based on the time
        long elapsed = System.currentTimeMillis() - startTime;
        gl.glRotatef(elapsed * (30f / 1000f), 0, 1, 0);
        gl.glRotatef(elapsed * (15f / 1000f), 1, 0, 0);

        // Draw the model
        cube.draw(gl);

        // Keep track of number of frames drawn
        numFrames++;
        long fpsElapsed = System.currentTimeMillis() - fpsStartTime;
        if (fpsElapsed > 5 * 1000) { // every 5 seconds
            float fps = (numFrames * 1000.0F) / fpsElapsed;
            Log.d(TAG, "Frames per second: " + fps + " (" + numFrames
                + " frames in " + fpsElapsed + " ms)");
            fpsStartTime = System.currentTimeMillis();
            numFrames = 0;
        }
    }
}

package com.app.Graphics3d;
import javax.microedition.khronos.egl.EGLConfig;
import javax.microedition.khronos.opengles.GL10;
import android.content.Context;
import android.opengl.GLSurfaceView;
import android.opengl.GLU;
import android.util.Log;

class MyGLRenderer implements GLSurfaceView.Renderer {

```

```

private static final String TAG = "GLRenderer";
private final Context context;
private final MyCube cube = new MyCube();
private long startTime;
private long fpsStartTime;
private long numFrames;
MyGLRenderer(Context context) {
    this.context = context;
}

public void onSurfaceCreated(GL10 gl, EGLConfig config)
{
    boolean SEE_THRU = true;
    startTime = System.currentTimeMillis();
    fpsStartTime = startTime;
    numFrames = 0;
    // Define the lighting
    float lightAmbient[] = new float[] { 0.2f, 0.2f, 0.2f, 1 };
    float lightDiffuse[] = new float[] { 1, 1, 1, 1 };
    float[] lightPos = new float[] { 1, 1, 1, 1 };
    gl.glEnable(GL10.GL_LIGHTING);
    gl.glEnable(GL10.GL_LIGHT0);
    gl.glLightfv(GL10.GL_LIGHT0, GL10.GL_AMBIENT, lightAmbient, 0);
    gl.glLightfv(GL10.GL_LIGHT0, GL10.GL_DIFFUSE, lightDiffuse, 0);
    gl.glLightfv(GL10.GL_LIGHT0, GL10.GL_POSITION, lightPos, 0);
    // What is the cube made of?
    float matAmbient[] = new float[] { 1, 1, 1, 1 };
    float matDiffuse[] = new float[] { 1, 1, 1, 1 };
    gl.glMaterialfv(GL10.GL_FRONT_AND_BACK, GL10.GL_AMBIENT,
        matAmbient, 0);
    gl.glMaterialfv(GL10.GL_FRONT_AND_BACK, GL10.GL_DIFFUSE,
        matDiffuse, 0);
    // Set up any OpenGL options we need
    gl.glEnable(GL10.GL_DEPTH_TEST);
    gl.glDepthFunc(GL10.GL_LEQUAL);
}

```

```

        gl.glEnableClientState(GL10.GL_VERTEX_ARRAY);
        if (SEE_THRU) {
            gl.glDisable(GL10.GL_DEPTH_TEST);
            gl.glEnable(GL10.GL_BLEND);
            gl.glBlendFunc(GL10.GL_SRC_ALPHA, GL10.GL_ONE);
        }
        // Enable textures
        gl.glEnableClientState(GL10.GL_TEXTURE_COORD_ARRAY);
        gl.glEnable(GL10.GL_TEXTURE_2D);

        // Load the cube's texture from a bitmap
        MyCube.loadTexture(gl, context, R.drawable.one);
    }

    public void onSurfaceChanged(GL10 gl, int width, int height)
    {
        // Define the view frustum
        gl.glViewport(0, 0, width, height);
        gl.glMatrixMode(GL10.GL_PROJECTION);
        gl.glLoadIdentity();
        float ratio = (float) width / height;
        GLU.gluPerspective(gl, 45.0f, ratio, 1, 100f);
    }

    public void onDrawFrame(GL10 gl) {
        // Clear the screen to black
        gl.glClear(GL10.GL_COLOR_BUFFER_BIT
        | GL10.GL_DEPTH_BUFFER_BIT);

        // Position model so we can see it
        gl.glMatrixMode(GL10.GL_MODELVIEW);
        gl.glLoadIdentity();
        gl.glTranslatef(0, 0, -3.0f);

        // Set rotation angle based on the time
        long elapsed = System.currentTimeMillis() - startTime;
        gl.glRotatef(elapsed * (30f / 1000f), 0, 1, 0);
    }

```

```

gl.glRotatef(elapsed * (15f / 1000f), 1, 0, 0);

// Draw the model
cube.draw(gl);
// Keep track of number of frames drawn
numFrames++;
long fpsElapsed = System.currentTimeMillis() - fpsStartTime;
if (fpsElapsed > 5 * 1000) { // every 5 seconds
float fps = (numFrames * 1000.0F) / fpsElapsed;
Log.d(TAG, "Frames per second: " + fps + " (" + numFrames
+ " frames in " + fpsElapsed + " ms)");
fpsStartTime = System.currentTimeMillis();
numFrames = 0;
}
}
}

```

6. Create MyCube.java, which will look like this:

```

package com.app.Graphics3d;
import java.nio.ByteBuffer;
import java.nio.ByteOrder;
import java.nio.IntBuffer;
import javax.microedition.khronos.opengles.GL10;
import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.opengl.GLUtils;
class MyCube {
    private final IntBuffer mVertexBuffer;
    private final IntBuffer mTextureBuffer;
    public MyCube() {
        int one = 65536;
        int half = one / 2;
        int vertices[] = {
            // FRONT

```

```

-half, -half, half, half, -half, half,
-half, half, half, half, half, half,
// BACK
-half, -half, -half, -half, half, -half,
half, -half, -half, half, half, -half,
// LEFT
-half, -half, half, -half, half, half,
-half, -half, -half, -half, half, -half,
// RIGHT
half, -half, -half, half, half, -half,
half, -half, half, half, half, half,
// TOP
-half, half, half, half, half, half,
-half, half, -half, half, half, -half,
// BOTTOM
-half, -half, half, -half, -half, -half,
half, -half, half, half, -half, -half, };

```

```

int texCoords[] = {
// FRONT
0, one, one, one, one, 0, 0, one, 0,
// BACK
one, one, one, 0, 0, one, 0, 0,
// LEFT
one, one, one, 0, 0, one, 0, 0,
// RIGHT
one, one, one, 0, 0, one, 0, 0,
// TOP
one, 0, 0, 0, one, one, 0, one,
// BOTTOM
0, 0, 0, one, one, 0, one, one, };

```

```

ByteBuffer vbb = ByteBuffer.allocateDirect(vertices.length * 4);
vbb.order(ByteOrder.nativeOrder());
mVertexBuffer = vbb.asIntBuffer();

```

```

        mVertexBuffer.put(vertices);
        mVertexBuffer.position(0);
        // ...
        ByteBuffer tbb = ByteBuffer.allocateDirect(texCoords.length * 4);
        tbb.order(ByteOrder.nativeOrder());
        mTextureBuffer = tbb.asIntBuffer();
        mTextureBuffer.put(texCoords);
        mTextureBuffer.position(0);
    }
    public void draw(GL10 gl) {
        gl.glVertexPointer(3, GL10.GL_FIXED, 0, mVertexBuffer);

        gl.glEnable(GL10.GL_TEXTURE_2D); // workaround bug 3623
        gl.glTexCoordPointer(2, GL10.GL_FIXED, 0, mTextureBuffer);

        gl.glColor4f(1, 1, 1, 1);
        gl.glNormal3f(0, 0, 1);
        gl.glDrawArrays(GL10.GL_TRIANGLE_STRIP, 0, 4);
        gl.glNormal3f(0, 0, -1);
        gl.glDrawArrays(GL10.GL_TRIANGLE_STRIP, 4, 4);

        gl.glColor4f(1, 1, 1, 1);
        gl.glNormal3f(-1, 0, 0);
        gl.glDrawArrays(GL10.GL_TRIANGLE_STRIP, 8, 4);
        gl.glNormal3f(1, 0, 0);
        gl.glDrawArrays(GL10.GL_TRIANGLE_STRIP, 12, 4);

        gl.glColor4f(1, 1, 1, 1);
        gl.glNormal3f(0, 1, 0);
        gl.glDrawArrays(GL10.GL_TRIANGLE_STRIP, 16, 4);
        gl.glNormal3f(0, -1, 0);
        gl.glDrawArrays(GL10.GL_TRIANGLE_STRIP, 20, 4);
    }

    static void loadTexture(GL10 gl, Context context, int resource) {
        Bitmap bmp =
            BitmapFactory.decodeResource(context.getResources(), resource);
    }

```

```

        GLUtils.texImage2D(GL10.GL_TEXTURE_2D, 0, bmp, 0);
        gl.glTexParameterx(GL10.GL_TEXTURE_2D,
        GL10.GL_TEXTURE_MIN_FILTER, GL10.GL_LINEAR);
        gl.glTexParameterx(GL10.GL_TEXTURE_2D,
        GL10.GL_TEXTURE_MAG_FILTER, GL10.GL_LINEAR);
        bmp.recycle();
    }
}

```

```

package com.app.Graphics3d;
import java.nio.ByteBuffer;
import java.nio.ByteOrder;
import java.nio.IntBuffer;
import javax.microedition.khronos.opengles.GL10;
import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.opengl.GLUtils;
class MyCube {
    private final IntBuffer mVertexBuffer;
    private final IntBuffer mTextureBuffer;
    public MyCube() {
        int one = 65536;
        int half = one / 2;
        int vertices[] = {
            // FRONT
            -half, -half, half, half, -half, half,
            -half, half, half, half, half, half,
            // BACK
            -half, -half, -half, -half, half, -half,
            half, -half, -half, half, half, -half,
            // LEFT
            -half, -half, half, -half, half, half,
            -half, -half, -half, -half, half, -half,
            // RIGHT

```



```

    half, -half, -half, half, half, -half,
    half, -half, half, half, half, half,
    // TOP
    -half, half, half, half, half, half,
    -half, half, -half, half, half, -half,
    // BOTTOM
    -half, -half, half, -half, -half, -half,
    half, -half, half, half, -half, -half,
};

int texCoords[] = {
    // FRONT
    0, one, one, one, 0, 0, one, 0,
    // BACK
    one, one, one, 0, 0, one, 0, 0,
    // LEFT
    one, one, one, 0, 0, one, 0, 0,
    // RIGHT
    one, one, one, 0, 0, one, 0, 0,
    // TOP
    one, 0, 0, 0, one, one, 0, one,
    // BOTTOM
    0, 0, 0, one, one, 0, one, one, };

ByteBuffer vbb = ByteBuffer.allocateDirect(vertices.length * 4);
vbb.order(ByteOrder.nativeOrder());
mVertexBuffer = vbb.asIntBuffer();
mVertexBuffer.put(vertices);
mVertexBuffer.position(0);

// ...
ByteBuffer tbb = ByteBuffer.allocateDirect(texCoords.length * 4);
tbb.order(ByteOrder.nativeOrder());
mTextureBuffer = tbb.asIntBuffer();
mTextureBuffer.put(texCoords);
mTextureBuffer.position(0);
}

```

```

public void draw(GL10 gl) {
    gl.glVertexPointer(3, GL10.GL_FIXED, 0, mVertexBuffer);

    gl.glEnable(GL10.GL_TEXTURE_2D); // workaround bug 3623
    gl.glTexCoordPointer(2, GL10.GL_FIXED, 0, mTextureBuffer);

    gl.glColor4f(1, 1, 1, 1);
    gl.glNormal3f(0, 0, 1);
    gl.glDrawArrays(GL10.GL_TRIANGLE_STRIP, 0, 4);
    gl.glNormal3f(0, 0, -1);
    gl.glDrawArrays(GL10.GL_TRIANGLE_STRIP, 4, 4);

    gl.glColor4f(1, 1, 1, 1);
    gl.glNormal3f(-1, 0, 0);
    gl.glDrawArrays(GL10.GL_TRIANGLE_STRIP, 8, 4);
    gl.glNormal3f(1, 0, 0);
    gl.glDrawArrays(GL10.GL_TRIANGLE_STRIP, 12, 4);

    gl.glColor4f(1, 1, 1, 1);
    gl.glNormal3f(0, 1, 0);
    gl.glDrawArrays(GL10.GL_TRIANGLE_STRIP, 16, 4);
    gl.glNormal3f(0, -1, 0);
    gl.glDrawArrays(GL10.GL_TRIANGLE_STRIP, 20, 4);
}

    static void loadTexture(GL10 gl, Context context, int resource) {
        Bitmap bmp =
            BitmapFactory.decodeResource(context.getResources(), resource);
        GLUtils.texImage2D(GL10.GL_TEXTURE_2D, 0, bmp, 0);
        gl.glTexParameterx(GL10.GL_TEXTURE_2D,
            GL10.GL_TEXTURE_MIN_FILTER, GL10.GL_LINEAR);
        gl.glTexParameterx(GL10.GL_TEXTURE_2D,
            GL10.GL_TEXTURE_MAG_FILTER, GL10.GL_LINEAR);
        bmp.recycle();
    }
}

```

Run the Application.

Steps to Create:

1.) Open Android Studio Use the New Project Wizard and select Android Project Give the respective project name i.e. Graphics3d. Enter following information:

- Project name: Graphics3d
- Build Target: Google APIs
- Application name: Graphics3d
- Package name: app. Com.app.Graphics3d
- Create Activity: Graphics3d
- On Clicking Finish Graphics3d code structure is generated with the necessary Android Packages being imported along with Graphics3d.java. Graphics3d class will look like following:

```
package com.app.Graphics3d;
import android.app.Activity;
import android.os.Bundle;
public class Graphics3d extends Activity {
    MyGLView view;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        view = new MyGLView(this);
        setContentView(view);
    }
    @Override
    protected void onPause() {
        super.onPause();
        view.onPause();
    }
    @Override
    protected void onResume() {
        super.onResume();
        view.onResume();
    }
}
```



Figure-18 3D Graphics Code Output

In this unit we learned how to build 3D Graphics using android Open-GL

Audio Player 8

Following is the example of implementing an audio player to play a song or audio with multiple playback options using MediaPlayer.

Create a new android application using android studio and give names as MediaPlayerExample. In case if you are not aware of creating an app in android studio check this article [Android Hello World App](#).

As discussed, create a new raw folder in res directory and add one music file like as shown below to play it by using MediaPlayer class.

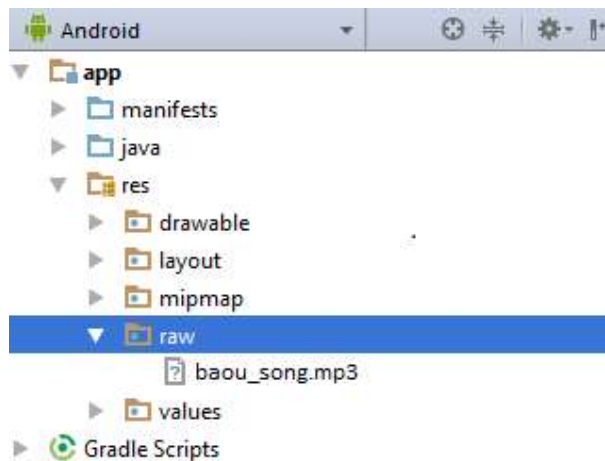


Figure-19 MediaPlayer Project structure and upload Music File (*.mp3)

Now open activity_main.xml file from \res\layout folder path and write the code like as shown below.

activity_main.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:paddingLeft="10dp"
    android:paddingRight="10dp">
```

```
<TextView
    android:id="@+id/txtVw1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Now Playing: "
    android:layout_marginTop="30dp"
    android:textAppearance="?android:attr/textAppearanceMedium" />
```

```
<TextView
    android:id="@+id/txtSname"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id/txtVw1"
    android:layout_toRightOf="@+id/txtVw1"
    android:text="TextView" />
```

```
<ImageView
    android:id="@+id/imgLogo"
    android:layout_width="match_parent"
    android:layout_height="450dp"
    android:layout_below="@+id/txtVw1"
    android:src="@drawable/baou" />
```

```
<ImageButton
    android:id="@+id/btnBackward"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_marginBottom="44dp"
    android:layout_marginLeft="20dp"
```

```

        android:src="@android:drawable/ic_media_rew" />
<ImageButton
    android:id="@+id/btnPlay"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignTop="@+id/btnBackward"
    android:layout_marginLeft="20dp"
    android:layout_toRightOf="@+id/btnBackward"
    android:src="@android:drawable/ic_media_play" />
<ImageButton
    android:id="@+id/btnPause"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignTop="@+id/btnPlay"
    android:layout_marginLeft="20dp"
    android:layout_toRightOf="@+id/btnPlay"
    android:src="@android:drawable/ic_media_pause" />
<ImageButton
    android:id="@+id/btnForward"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignTop="@+id/btnPause"
    android:layout_marginLeft="20dp"
    android:layout_toRightOf="@+id/btnPause"
    android:contentDescription="@+id/imageButton3"
    android:src="@android:drawable/ic_media_ff" />
<TextView

```

```

        android:id="@+id/txtStartTime"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignTop="@+id/sBar"
        android:text="0 min, 0 sec" />
<SeekBar
    android:id="@+id/sBar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_above="@+id/btnBackward"
    android:layout_toLeftOf="@+id/txtSongTime"
    android:layout_toRightOf="@+id/txtStartTime" />
<TextView
    android:id="@+id/txtSongTime"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_toRightOf="@+id/btnForward"
    android:layout_alignTop="@+id/sBar"
    android:text="0 min, 0 sec " />
</RelativeLayout>

```

Now open your main activity file MainActivity.java from \java\com.baou.audioplayerexample path and write the code like as shown below.

MainActivity.java

```

package com.baou.mediaplayerexample;

import android.media.MediaPlayer;

import android.os.Handler;

```



```

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.ImageButton;

import android.widget.SeekBar;

import android.widget.TextView;

import android.widget.Toast;

import java.util.concurrent.TimeUnit;

public class MainActivity extends AppCompatActivity {

    private ImageButton forwardbtn, backwardbtn, pausebtn, playbtn;

    private MediaPlayer mPlayer;

    private TextView songName, startTime, songTime;

    private SeekBar songPrgs;

    private static int oTime =0, sTime =0, eTime =0, fTime = 5000, bTime = 5000;

    private Handler hdlr = new Handler();

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        backwardbtn = (ImageButton)findViewById(R.id.btnBackward);

        forwardbtn = (ImageButton)findViewById(R.id.btnForward);

        playbtn = (ImageButton)findViewById(R.id.btnPlay);

        pausebtn = (ImageButton)findViewById(R.id.btnPause);

        songName = (TextView)findViewById(R.id.txtSname);

        startTime = (TextView)findViewById(R.id.txtStartTime);

        songTime = (TextView)findViewById(R.id.txtSongTime);

        songName.setText("Baitikochi Chuste");

```

```

mPlayer = MediaPlayer.create(this, R.raw.baitikochi_chuste);
songPrgs = (SeekBar)findViewById(R.id.sBar);
songPrgs.setClickable(false);
pausebtn.setEnabled(false);
playbtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v)
    Toast.makeText(MainActivity.this, "Playing Audio",
    Toast.LENGTH_SHORT).show();

    mPlayer.start();

    eTime = mPlayer.getDuration();

    sTime = mPlayer.getCurrentPosition();

    if(oTime == 0){
        songPrgs.setMax(eTime);
        oTime =1;
    }

    songTime.setText(String.format("%d min, %d sec",
    TimeUnit.MILLISECONDS.toMinutes(eTime),
        TimeUnit.MILLISECONDS.toSeconds(eTime) -
    TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.toMinutes(eTime))) );

    startTime.setText(String.format("%d min, %d sec",
    TimeUnit.MILLISECONDS.toMinutes(sTime),
        TimeUnit.MILLISECONDS.toSeconds(sTime) -
    TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.toMinutes(sTime))) );

    songPrgs.setProgress(sTime);

    hdlr.postDelayed(UpdateSongTime, 100);

    pausebtn.setEnabled(true);

    playbtn.setEnabled(false);
}

```

```

});

pausebtn.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View v) {

        mPlayer.pause();

        pausebtn.setEnabled(false);

        playbtn.setEnabled(true);

        Toast.makeText(getApplicationContext(),"Pausing Audio",
Toast.LENGTH_SHORT).show();

    }

});

forwardbtn.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View v) {

        if((sTime + fTime) <= eTime)

        {

            sTime = sTime + fTime;

            mPlayer.seekTo(sTime);

        }

        else

        {

            Toast.makeText(getApplicationContext(), "Cannot jump forward 5
seconds", Toast.LENGTH_SHORT).show();

        }

        if(!playbtn.isEnabled()){

            playbtn.setEnabled(true);

        }

    }

}

```

```

});

backwardbtn.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View v) {

        if((sTime - bTime) > 0)

        {

            sTime = sTime - bTime;

            mPlayer.seekTo(sTime);

        }

        else

        {

            Toast.makeText(getApplicationContext(), "Cannot jump backward 5
seconds", Toast.LENGTH_SHORT).show();

        }

        if(!playbtn.isEnabled()){

            playbtn.setEnabled(true);

        }

    }

});

}

private Runnable UpdateSongTime = new Runnable() {

    @Override

    public void run() {

        sTime = mPlayer.getCurrentPosition();

        startTime.setText(String.format("%d min, %d sec",
TimeUnit.MILLISECONDS.toMinutes(sTime),

            TimeUnit.MILLISECONDS.toSeconds(sTime)
TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.toMinutes(sTime))) );

```

```

        songPrgs.setProgress(sTime);
        hdlr.postDelayed(this, 100);
    }
};
}

```

If you observe above code we used MediaPlayer object properties to play, pause song and changing the song position either forward or backward based on our requirements.

Output of Android Audio Player Example

When we run above program in android studio, we will get the result like when we click on play button song will start play and it will show the song duration details. If we click on pause button, the song will stop playing and use forward and backward buttons to move song forward or backward based on your requirements.



Media Player apps code output

If you observe above result, This is how we can implement audio player app in android applications with multiple playback options based on our requirements.

Animation

9

Let's create android animation application, open android studio and Execute project creation wizard with own credentials,

Loading Animation when UI widget is clicked: Our aim is to show an animation when any widget(lets say TextView) is clicked. For that we need to use the Animation Class. The xml file that contains the animation logic is loaded using AnimationUtils class by calling the loadAnimation() function. The below snippet shows this implementation.

```
Animation animation;  
  
animation = AnimationUtils.loadAnimation(getApplicationContext(),  
  
R.anim.sample_animation);
```

To start the animation we need to call the startAnimation() function on the UI element as shown in the snippet below:

```
sampleTextView.startAnimation(animation);
```

Here we perform the animation on a textview component by passing the type of Animation as the parameter.

Setting the Animation Listeners

This is only needed if we wish to listen to events like start, end or repeat. For this the activity must implement AnimationListener and the following methods need to overridden.

onAnimationStart : This will be triggered once the animation started

onAnimationEnd : This will be triggered once the animation is over

onAnimationRepeat : This will be triggered if the animation repeats

Android Animation Project Structure

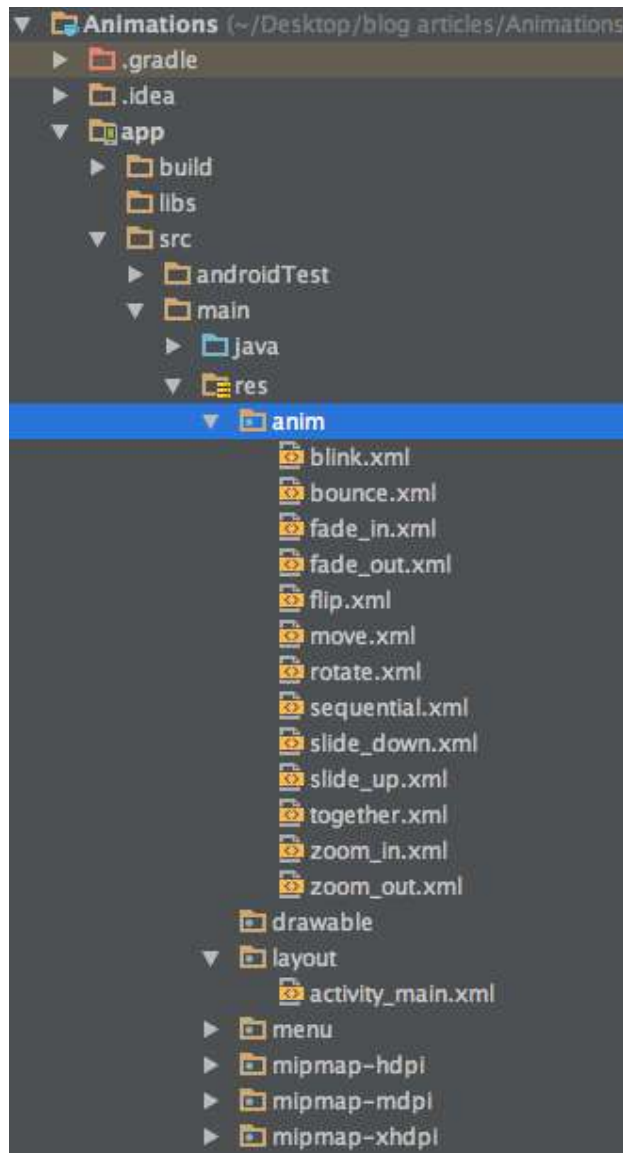


Figure-22 Project Structure for Android Animation

As you can see, we've included the xml of all the major types of animations covered above.

Android Animation Examples XML Code

Here all sample code for most of the common android animations are available.

Fade In Animation : fade_in.xml

```
<set xmlns:android="http://schemas.android.com/apk/res/android"  
    android:fillAfter="true" >
```

```

<alpha
    android:duration="1000"
    android:fromAlpha="0.0"
    android:interpolator="@android:anim/accelerate_interpolator"
    android:toAlpha="1.0" />
</set>

```

Here alpha references the opacity of an object. An object with lower alpha values is more transparent, while an object with higher alpha values is less transparent, more opaque. Fade in animation is nothing but increasing alpha value from 0 to 1.

Fade Out Animation: fade_out.xml

```

<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:fillAfter="true" >
    <alpha
        android:duration="1000"
        android:fromAlpha="1.0"
        android:interpolator="@android:anim/accelerate_interpolator"
        android:toAlpha="0.0" />
</set>

```

Fade out android animation is exactly opposite to fade in, where we need to decrease the alpha value from 1 to 0.

Cross Fading Animation

Cross fading is performing fade in animation on one TextView while other TextView is fading out. This can be done by using fade_in.xml and fade_out.xml on the two TextViews. The code will be discussed in the MainActivity.java

Blink Animation : blink.xml

```

<set xmlns:android="http://schemas.android.com/apk/res/android">
    <alpha android:fromAlpha="0.0"
        android:toAlpha="1.0"
        android:interpolator="@android:anim/accelerate_interpolator"
        android:duration="600"
        android:repeatMode="reverse"

```



```
        android:repeatCount="infinite"/>
</set>
```

Here fade in and fade out are performed infinitely in reverse mode each time.

Zoom In Animation: zoom_in.xml

```
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:fillAfter="true" >
    <scale
        xmlns:android="http://schemas.android.com/apk/res/android"
        android:duration="1000"
        android:fromXScale="1"
        android:fromYScale="1"
        android:pivotX="50%"
        android:pivotY="50%"
        android:toXScale="3"
        android:toYScale="3" >
    </scale>
</set>
```

We use pivotX="50%" and pivotY="50%" to perform zoom from the center of the element.

Zoom Out Animation: zoom_out.xml

```
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:fillAfter="true" >
    <scale
        xmlns:android="http://schemas.android.com/apk/res/android"
        android:duration="1000"
        android:fromXScale="1.0"
        android:fromYScale="1.0"
        android:pivotX="50%"
        android:pivotY="50%"
        android:toXScale="0.5"
        android:toYScale="0.5" >
    </scale>
</set>
```

Notice that android:from and android:to are opposite in zoom_in.xml and zoom_out.xml.

Rotate Animation: rotate.xml

```
<set xmlns:android="http://schemas.android.com/apk/res/android">
  <rotate android:fromDegrees="0"
    android:toDegrees="360"
    android:pivotX="50%"
    android:pivotY="50%"
    android:duration="600"
    android:repeatMode="restart"
    android:repeatCount="infinite"
    android:interpolator="@android:anim/cycle_interpolator"/>
</set>
```

A from/toDegrees tag is used here to specify the degrees and a cyclic interpolator is used.

Move Animation: move.xml

```
<set
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:interpolator="@android:anim/linear_interpolator"
  android:fillAfter="true">
  <translate
    android:fromXDelta="0%p"
    android:toXDelta="75%p"
    android:duration="800" />
</set>
```

Slide Up Animation: slide_up.xml

```
<set xmlns:android="http://schemas.android.com/apk/res/android"
  android:fillAfter="true" >
  <scale
    android:duration="500"
    android:fromXScale="1.0"
    android:fromYScale="1.0"
```

```
        android:interpolator="@android:anim/linear_interpolator"
        android:toXScale="1.0"
        android:toYScale="0.0" />
</set>
```

It's achieved by setting `android:fromYScale="1.0"` and `android:toYScale="0.0"` inside the scale tag.

Slide Down Animation: slide_down.xml

```
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:fillAfter="true">
    <scale
        android:duration="500"
        android:fromXScale="1.0"
        android:fromYScale="0.0"
        android:toXScale="1.0"
        android:toYScale="1.0" />
</set>
```

This is just the opposite of `slide_up.xml`.

Bounce Animation: bounce.xml

```
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:fillAfter="true"
    android:interpolator="@android:anim/bounce_interpolator">
    <scale
        android:duration="500"
        android:fromXScale="1.0"
        android:fromYScale="0.0"
        android:toXScale="1.0"
        android:toYScale="1.0" />
</set>
```

Here bounce interpolator is used to complete the animation in bouncing fashion.

Sequential Animation: sequential.xml

```

<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:fillAfter="true"
    android:interpolator="@android:anim/linear_interpolator" >
<!-- Move -->
<translate
    android:duration="800"
    android:fillAfter="true"
    android:fromXDelta="0%p"
    android:startOffset="300"
    android:toXDelta="75%p" />
<translate
    android:duration="800"
    android:fillAfter="true"
    android:fromYDelta="0%p"
    android:startOffset="1100"
    android:toYDelta="70%p" />
<translate
    android:duration="800"
    android:fillAfter="true"
    android:fromXDelta="0%p"
    android:startOffset="1900"
    android:toXDelta="-75%p" />
<translate
    android:duration="800"
    android:fillAfter="true"
    android:fromYDelta="0%p"
    android:startOffset="2700"
    android:toYDelta="-70%p" />

<!-- Rotate 360 degrees -->
<rotate
    android:duration="1000"
    android:fromDegrees="0"
    android:interpolator="@android:anim/cycle_interpolator"
    android:pivotX="50%"
    android:pivotY="50%"

```

```
        android:startOffset="3800"
        android:repeatCount="infinite"
        android:repeatMode="restart"
        android:toDegrees="360" />
</set>
```

Here a different android:startOffset is used from the transitions to keep them sequential.

Together Animation: together.xml

```
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:fillAfter="true"
    android:interpolator="@android:anim/linear_interpolator" >

    <!-- Move -->
    <scale
        xmlns:android="http://schemas.android.com/apk/res/android"
        android:duration="4000"
        android:fromXScale="1"
        android:fromYScale="1"
        android:pivotX="50%"
        android:pivotY="50%"
        android:toXScale="4"
        android:toYScale="4" >
    </scale>

    <!-- Rotate 180 degrees -->
    <rotate
        android:duration="500"
        android:fromDegrees="0"
        android:pivotX="50%"
        android:pivotY="50%"
        android:repeatCount="infinite"
        android:repeatMode="restart"

        android:toDegrees="360" />
</set>
```

Here android:startOffset is removed to let them happen simultaneously.

Code : The activity_main.xml layout consists of a ScrollView and RelativeLayout (we'll discuss this in a later) in which every animation type is invoked on the text using their respective buttons. The xml file is shown below :

activity_main.xml

```
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <Button
            android:id="@+id/btnFadeIn"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="5dp"
            android:text="Fade In" />
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceMedium"
            android:text="Fade In"
            android:id="@+id/txt_fade_in"
            android:layout_alignBottom="@+id/btnFadeIn"
            android:layout_alignLeft="@+id/txt_fade_out"
            android:layout_alignStart="@+id/txt_fade_out" />
        <Button
```

```
android:id="@+id/btnFadeOut"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_margin="5dp"  
android:layout_below="@id/btnFadeIn"  
android:text="Fade Out" />
```

<Button

```
android:id="@+id/btnCrossFade"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_margin="5dp"  
android:layout_below="@id/btnFadeOut"  
android:text="Cross Fade" />
```

<TextView

```
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:textAppearance="?android:attr/textAppearanceMedium"  
android:text="Cross Fade In"  
android:id="@+id/txt_out"  
android:visibility="gone"  
android:layout_gravity="center_horizontal"  
android:layout_alignTop="@+id/txt_in"  
android:layout_alignLeft="@+id/txt_in"  
android:layout_alignStart="@+id/txt_in" />
```

<Button

```
android:id="@+id/btnBlink"  
android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:layout_below="@id/btnCrossFade"
    android:text="Blink" />
```

```
<Button
```

```
    android:id="@+id/btnZoomIn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:layout_below="@id/btnBlink"
    android:text="Zoom In" />
```

```
<TextView
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:text="Blink"
    android:id="@+id/txt_blink"
    android:layout_gravity="center_horizontal"
    android:layout_alignBottom="@+id/btnBlink"
    android:layout_alignLeft="@+id/txt_zoom_in"
    android:layout_alignStart="@+id/txt_zoom_in" />
```

```
<Button
```

```
    android:id="@+id/btnZoomOut"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:layout_below="@id/btnZoomIn"
```



```
    android:text="Zoom Out" />
```

```
<Button
```

```
    android:id="@+id/btnRotate"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:layout_below="@id/btnZoomOut"
    android:text="Rotate" />
```

```
<Button
```

```
    android:id="@+id/btnMove"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:layout_below="@id/btnRotate"
    android:text="Move" />
```

```
<Button
```

```
    android:id="@+id/btnSlideUp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:layout_below="@id/btnMove"
    android:text="Slide Up" />
```

```
<TextView
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:text="Fade Out"
```

```
    android:id="@+id/txt_fade_out"
    android:layout_gravity="center_horizontal"
    android:layout_alignBottom="@+id/btnFadeOut"
    android:layout_alignLeft="@+id/txt_in"
    android:layout_alignStart="@+id/txt_in" />
```

```
<Button
```

```
    android:id="@+id/btnSlideDown"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:layout_below="@id/btnSlideUp"
    android:text="Slide Down" />
```

```
<Button
```

```
    android:id="@+id/btnBounce"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:layout_below="@id/btnSlideDown"
    android:text="Bounce" />
```

```
<Button
```

```
    android:id="@+id/btnSequential"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:layout_below="@id/btnBounce"
    android:text="Sequential Animation" />
```

```
<Button
```

```
android:id="@+id/btnTogether"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_below="@id/btnSequential"  
android:layout_margin="5dp"  
android:text="Together Animation" />
```

```
<TextView
```

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:textAppearance="?android:attr/textAppearanceMedium"  
    android:text="Cross Fade Out"  
    android:id="@+id/txt_in"  
    android:layout_gravity="center_horizontal"  
    android:layout_alignBottom="@+id/btnCrossFade"  
    android:layout_alignLeft="@+id/txt_blink"  
    android:layout_alignStart="@+id/txt_blink" />
```

```
<TextView
```

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:textAppearance="?android:attr/textAppearanceMedium"  
    android:text="Zoom In"  
    android:id="@+id/txt_zoom_in"  
    android:layout_alignBottom="@+id/btnZoomIn"  
    android:layout_alignLeft="@+id/txt_zoom_out"  
    android:layout_alignStart="@+id/txt_zoom_out" />
```

```
<TextView
```

```
    android:layout_width="wrap_content"
```

```

        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="Zoom Out"
        android:id="@+id/txt_zoom_out"
        android:layout_alignBottom="@+id/btnZoomOut"
        android:layout_toRightOf="@+id/btnSequential"
        android:layout_toEndOf="@+id/btnSequential" />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:text="Rotate"
    android:id="@+id/txt_rotate"
    android:layout_above="@+id/btnMove"
    android:layout_toRightOf="@+id/btnSequential"
    android:layout_toEndOf="@+id/btnSequential" />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:text="Move"
    android:id="@+id/txt_move"
    android:layout_alignBottom="@+id/btnMove"
    android:layout_alignLeft="@+id/txt_slide_up"
    android:layout_alignStart="@+id/txt_slide_up" />
<TextView
    android:layout_width="wrap_content"

```

```
android:layout_height="wrap_content"
android:textAppearance="?android:attr/textAppearanceMedium"
android:text="Slide Up"
android:id="@+id/txt_slide_up"
android:layout_alignBottom="@+id/btnSlideUp"
android:layout_toRightOf="@+id/btnSequential"
android:layout_toEndOf="@+id/btnSequential" />
```

```
<TextView
```

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:textAppearance="?android:attr/textAppearanceMedium"
android:text="Slide Down"
android:id="@+id/txt_slide_down"
android:layout_alignBottom="@+id/btnSlideDown"
android:layout_alignLeft="@+id/txt_slide_up"
android:layout_alignStart="@+id/txt_slide_up" />
```

```
<TextView
```

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:textAppearance="?android:attr/textAppearanceMedium"
android:text="Bounce"
android:id="@+id/txt_bounce"
android:layout_alignBottom="@+id/btnBounce"
android:layout_alignLeft="@+id/txt_slide_down"
android:layout_alignStart="@+id/txt_slide_down" />
```

```
<TextView
```

```
android:layout_width="wrap_content"
```

```

        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="Sequential"
        android:id="@+id/txt_seq"
        android:layout_alignBottom="@+id/btnSequential"
        android:layout_alignLeft="@+id/txt_bounce"
        android:layout_alignStart="@+id/txt_bounce" />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:text="Together"
    android:id="@+id/txt_tog"
    android:layout_alignBottom="@+id/btnTogether"
    android:layout_toRightOf="@+id/btnSequential"
    android:layout_toEndOf="@+id/btnSequential" />
</RelativeLayout>
</ScrollView>

```

To sum up, a RelativeLayout, as the name suggests the arrangement of UI Components is relative to each other.

The MainActivity.java file contains the onClick Listeners for every button related to its animation type. It's source code is given below.

```

package com.baou.animations;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;

```

```

import android.view.View;

import android.view.animation.Animation;

import android.view.animation.AnimationUtils;

import android.widget.Button;

import android.widget.TextView;

public class MainActivity extends Activity {

    Button btnFadeIn, btnFadeOut, btnCrossFade, btnBlink, btnZoomIn,
        btnZoomOut, btnRotate, btnMove, btnSlideUp, btnSlideDown,
        btnBounce, btnSequential, btnTogether;

    Animation
    animFadeIn, animFadeOut, animBlink, animZoomIn, animZoomOut, animRotate
    , animMove, animSlideUp, animSlideDown, animBounce, animSequential, animTogether
    , animCrossFadeIn, animCrossFadeOut;

    TextView
    txtFadeIn, txtFadeOut, txtBlink, txtZoomIn, txtZoomOut, txtRotate, txtMove, txtSlideUp,
        txtSlideDown, txtBounce, txtSeq, txtTog, txtIn, txtOut;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        btnFadeIn = (Button) findViewById(R.id.btnFadeIn);

        btnFadeOut = (Button) findViewById(R.id.btnFadeOut);

        btnCrossFade = (Button) findViewById(R.id.btnCrossFade);

        btnBlink = (Button) findViewById(R.id.btnBlink);

        btnZoomIn = (Button) findViewById(R.id.btnZoomIn);

        btnZoomOut = (Button) findViewById(R.id.btnZoomOut);

        btnRotate = (Button) findViewById(R.id.btnRotate);

        btnMove = (Button) findViewById(R.id.btnMove);

```

```

btnSlideUp = (Button) findViewById(R.id.btnSlideUp);
btnSlideDown = (Button) findViewById(R.id.btnSlideDown);
btnBounce = (Button) findViewById(R.id.btnBounce);
btnSequential = (Button) findViewById(R.id.btnSequential);
btnTogether = (Button) findViewById(R.id.btnTogether);
txtFadeIn=(TextView)findViewById(R.id.txt_fade_in);
txtFadeOut=(TextView)findViewById(R.id.txt_fade_out);
txtBlink=(TextView)findViewById(R.id.txt_blink);
txtZoomIn=(TextView)findViewById(R.id.txt_zoom_in);
txtZoomOut=(TextView)findViewById(R.id.txt_zoom_out);
txtRotate=(TextView)findViewById(R.id.txt_rotate);
txtMove=(TextView)findViewById(R.id.txt_move);
txtSlideUp=(TextView)findViewById(R.id.txt_slide_up);
txtSlideDown=(TextView)findViewById(R.id.txt_slide_down);
txtBounce=(TextView)findViewById(R.id.txt_bounce);
txtSeq=(TextView)findViewById(R.id.txt_seq);
txtTog=(TextView)findViewById(R.id.txt_tog);
txtIn=(TextView)findViewById(R.id.txt_in);
txtOut=(TextView)findViewById(R.id.txt_out);
animFadeIn = AnimationUtils.loadAnimation(getApplicationContext(),
    R.anim.fade_in);
animFadeIn = AnimationUtils.loadAnimation(getApplicationContext(),
    R.anim.fade_in);
// fade in
btnFadeIn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

```



```

        txtFadeIn.setVisibility(View.VISIBLE);
        txtFadeIn.startAnimation(animFadeIn);
    }
});

animFadeOut = AnimationUtils.loadAnimation(getApplicationContext(),
    R.anim.fade_out);
// fade out
btnFadeOut.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        txtFadeOut.setVisibility(View.VISIBLE);
        txtFadeOut.startAnimation(animFadeOut);
    }
});

animCrossFadeIn = AnimationUtils.loadAnimation(getApplicationContext(),
    R.anim.fade_in);
animCrossFadeOut = AnimationUtils.loadAnimation(getApplicationContext(),
    R.anim.fade_out);
// cross fade
btnCrossFade.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        txtOut.setVisibility(View.VISIBLE);
        // start fade in animation
        txtOut.startAnimation(animCrossFadeIn);
        // start fade out animation

```

```

        txtIn.startAnimation(animCrossFadeOut);
    }
});
animBlink = AnimationUtils.loadAnimation(getApplicationContext(),
    R.anim.blink);
// blink
btnBlink.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        txtBlink.setVisibility(View.VISIBLE);
        txtBlink.startAnimation(animBlink);
    }
});
animZoomIn = AnimationUtils.loadAnimation(getApplicationContext(),
    R.anim.zoom_in);
// Zoom In
btnZoomIn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        txtZoomIn.setVisibility(View.VISIBLE);
        txtZoomIn.startAnimation(animZoomIn);
    }
});
animZoomOut = AnimationUtils.loadAnimation(getApplicationContext(),
    R.anim.zoom_out);
// Zoom Out
btnZoomOut.setOnClickListener(new View.OnClickListener() {

```

```

@Override
public void onClick(View v) {
    txtZoomOut.setVisibility(View.VISIBLE);
    txtZoomOut.startAnimation(animZoomOut);
}
});
animRotate = AnimationUtils.loadAnimation(getApplicationContext(),
    R.anim.rotate);
// Rotate
btnRotate.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        txtRotate.startAnimation(animRotate);
    }
});
animMove = AnimationUtils.loadAnimation(getApplicationContext(),
    R.anim.move);
// Move
btnMove.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        txtMove.startAnimation(animMove);
    }
});
animSlideUp = AnimationUtils.loadAnimation(getApplicationContext(),
    R.anim.slide_up);
// Slide Up

```

```

btnSlideUp.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        txtSlideUp.startAnimation(animSlideUp);
    }
});

animSlideDown = AnimationUtils.loadAnimation(getApplicationContext(),
    R.anim.slide_down);

// Slide Down

btnSlideDown.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        txtSlideDown.startAnimation(animSlideDown);
    }
});

animBounce = AnimationUtils.loadAnimation(getApplicationContext(),
    R.anim.bounce);

// Slide Down

btnBounce.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        txtBounce.startAnimation(animBounce);
    }
});

animSequential = AnimationUtils.loadAnimation(getApplicationContext(),
    R.anim.sequential);

// Sequential

```

```

btnSequential.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        txtSeq.startAnimation(animSequential);
    }
});
animTogether = AnimationUtils.loadAnimation(getApplicationContext(),
    R.anim.together);
// Together
btnTogether.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        txtTog.startAnimation(animTogether);
    }
});
}
}

```

As discussed before each textView animation is started by invoking the respective animation object in which the animation logic is loaded by AnimationUtils.loadAnimation() method. The crossFade animation consists of two TextViews in which one fades out and the other fades in.

The below example demonstrates the use of HttpURLConnection class. It creates a basic application that allows you to download HTML from a given web page.

To experiment with this example, you need to run this on an actual device on which wifi internet is connected.

Steps Description

1. You will use Android studio IDE to create an Android application under a package com.baou.myapplication.
2. Modify src/MainActivity.java file to add Activity code.
3. Modify layout XML file res/layout/activity_main.xml add any GUI component if required.
4. Modify AndroidManifest.xml to add necessary permissions.
5. Run the application and choose a running android device and install the application on it and verify the results.
6. Here is the content of src/MainActivity.java.

```
package com.baou.myapplication;

import android.app.ProgressDialog;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.net.ConnectivityManager;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.support.v7.app.ActionBarActivity;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
```

```

import android.widget.Toast;

import java.io.IOException;

import java.io.InputStream;

import java.net.HttpURLConnection;

import java.net.MalformedURLException;

import java.net.URL;

import java.net.URLConnection;

public class MainActivity extends ActionBarActivity {

    private ProgressDialog progressDialog;

    private Bitmap bitmap = null;

    Button b1;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        b1 = (Button) findViewById(R.id.button);

        b1.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View v) {

                checkInternetConenction();

                downloadImage("http://www.baou.edu.in/assets/images/baou_logo_n.png ");

            }

        });

    }

    private void downloadImage(String urlStr) {

        progressDialog = ProgressDialog.show(this, "", "Downloading Image from " + urlStr);

```

```

        final String url = urlStr;
new Thread() {
    public void run() {
        InputStream in = null;
        Message msg = Message.obtain();
        msg.what = 1;
        try {
            in = openHttpConnection(url);
            bitmap = BitmapFactory.decodeStream(in);
            Bundle b = new Bundle();
            b.putParcelable("bitmap", bitmap);
            msg.setData(b);
            in.close();
        } catch (IOException e1) {
            e1.printStackTrace();
        }
        messageHandler.sendMessage(msg);
    }
}.start();
}

private InputStream openHttpConnection(String urlStr) {
    InputStream in = null;
    int resCode = -1;
    try {
        URL url = new URL(urlStr);
        URLConnection urlConn = url.openConnection();
        if (!(urlConn instanceof HttpURLConnection)) {

```



```

        throw new IOException("URL is not an Http URL");
    }
    HttpURLConnection httpConn = (HttpURLConnection) urlConn;
    httpConn.setAllowUserInteraction(false);
    httpConn.setInstanceFollowRedirects(true);
    httpConn.setRequestMethod("GET");
    httpConn.connect();
    resCode = httpConn.getResponseCode();

    if (resCode == HttpURLConnection.HTTP_OK) {
        in = httpConn.getInputStream();
    }
} catch (MalformedURLException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
return in;
}

private Handler messageHandler = new Handler() {
    public void handleMessage(Message msg) {
        super.handleMessage(msg);
        ImageView img = (ImageView) findViewById(R.id.imageView);
        img.setImageBitmap((Bitmap) (msg.getData().getParcelable("bitmap")));
        progressDialog.dismiss();
    }
};

```

```

private boolean checkInternetConenction() {
    // get Connectivity Manager object to check connection
    ConnectivityManager connec
=
    (ConnectivityManager)getSystemService(getBaseContext().CONNECTIVITY_SERVI
    CE);
    // Check for network connections
    if ( connec.getNetworkInfo(0).getState() ==
        android.net.NetworkInfo.State.CONNECTED ||
        connec.getNetworkInfo(0).getState() ==
        android.net.NetworkInfo.State.CONNECTING ||
        connec.getNetworkInfo(1).getState() ==
        android.net.NetworkInfo.State.CONNECTING ||
        connec.getNetworkInfo(1).getState() ==
        android.net.NetworkInfo.State.CONNECTED ) {
        Toast.makeText(this, " Connected ", Toast.LENGTH_LONG).show();
        return true;
    }else if (
        connec.getNetworkInfo(0).getState() ==
        android.net.NetworkInfo.State.DISCONNECTED ||
        connec.getNetworkInfo(1).getState() ==
        android.net.NetworkInfo.State.DISCONNECTED ) {
        Toast.makeText(this, " Not Connected ", Toast.LENGTH_LONG).show();
        return false;
    }
    return false;
}
}

```

Here is the content of 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:paddingLeft="@dimen/activity_horizontal_margin"

    android:paddingRight="@dimen/activity_horizontal_margin"

    android:paddingTop="@dimen/activity_vertical_margin"

    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context=".MainActivity">

    <TextView

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:text="UI Animator Viewer"

        android:id="@+id/textView"

        android:textSize="25sp"

        android:layout_centerHorizontal="true" />

    <TextView

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:text="BAOU Network"

        android:id="@+id/textView2"

        android:layout_below="@+id/textView"

        android:layout_alignRight="@+id/textView"

        android:layout_alignEnd="@+id/textView"

        android:textColor="#ff36ff15"

        android:textIsSelectable="false"
```

```

        android:textSize="35dp" />
<ImageView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/imageView"
    android:layout_below="@+id/textView2"
    android:layout_centerHorizontal="true" />
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button"
    android:id="@+id/button"
    android:layout_below="@+id/imageView"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="76dp" />
</RelativeLayout>

```

Here is the content of Strings.xml.

```

<resources>
    <string name="app_name">My Application</string>
</resources>

```

Here is the content of AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.baou.myapplication" >
    <uses-permission android:name="android.permission.INTERNET">
        </uses-permission>

```

```

        <uses-permission android:name=
"android.permission.ACCESS_NETWORK_STATE">
        </uses-permission>
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
    <activity
        android:name=".MainActivity"
        android:label="@string/app_name" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>
</manifest>

```

Let's try to run your application. I assume you have connected your actual Android Mobile device with your computer. To run the app from android studio, open one of your project's activity files and click Run Android studio Run Icon icon from the tool bar. Before starting your application, Android studio will display following window to select an option where you want to run your Android application.

Select your mobile device as an option and then check your mobile device which will display following screen Now just click on button, it will check internet connection as well as it will download image.

Here is an example demonstrating the use of WebView Layout. It creates a basic web application that will ask you to specify a url and will load this url website in the WebView.

To experiment with this example, you need to run this on an actual device on which internet is running.

1. You will use Android studio to create an Android application under a package com.web.baou.myapplication.
2. Modify src/MainActivity.java file to add WebView code.
3. Modify the res/layout/activity_main to add respective XML components
4. Modify the AndroidManifest.xml to add the necessary permissions
5. Run the application and choose a running android device and install the application on it and verify the results.

Following is the content of the modified main activity file src/MainActivity.java.

```
package com.web.baou.myapplication;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends Activity {
    Button b1;
    EditText ed1;

    private WebView wv1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
```

```

b1=(Button)findViewById(R.id.button);
ed1=(EditText)findViewById(R.id.editText);

wv1=(WebView)findViewById(R.id.webView);
wv1.setWebViewClient(new MyBrowser());

b1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String url = ed1.getText().toString();

        wv1.getSettings().setLoadsImagesAutomatically(true);
        wv1.getSettings().setJavaScriptEnabled(true);
        wv1.setScrollBarStyle(View.SCROLLBARS_INSIDE_OVERLAY);
        wv1.loadUrl(url);
    }
});
}

private class MyBrowser extends WebViewClient {
    @Override
    public boolean shouldOverrideUrlLoading(WebView view, String url) {
        view.loadUrl(url);
        return true;
    }
}
}

```

Following is the modified content of the xml res/layout/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:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"

```

```
    android:paddingBottom="@dimen/activity_vertical_margin"  
tools:context=".MainActivity">
```

```
<TextView android:text="WebView" android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:id="@+id/textview"  
    android:textSize="35dp"  
    android:layout_alignParentTop="true"  
    android:layout_centerHorizontal="true" />
```

```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="BAOU"  
    android:id="@+id/textView"  
    android:layout_below="@+id/textview"  
    android:layout_centerHorizontal="true"  
    android:textColor="#ff7aff24"  
    android:textSize="35dp" />
```

```
<EditText  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:id="@+id/editText"  
    android:hint="Enter Text"  
    android:focusable="true"  
    android:textColorHighlight="#ff7eff15"  
    android:textColorHint="#ffff25e6"  
    android:layout_marginTop="46dp"  
    android:layout_below="@+id/imageView"  
    android:layout_alignParentLeft="true"  
    android:layout_alignParentStart="true"  
    android:layout_alignRight="@+id/imageView"  
    android:layout_alignEnd="@+id/imageView" />
```

```
<ImageView  
    android:layout_width="wrap_content"
```



```
android:layout_height="wrap_content"
android:id="@+id/imageView"
android:src="@drawable/abc"
android:layout_below="@+id/textView"
android:layout_centerHorizontal="true" />
```

<Button

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Enter"
android:id="@+id/button"
android:layout_alignTop="@+id/editText"
android:layout_toRightOf="@+id/imageView"
android:layout_toEndOf="@+id/imageView" />
```

<WebView

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/webView"
android:layout_below="@+id/button"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
android:layout_alignParentRight="true"
android:layout_alignParentEnd="true"
android:layout_alignParentBottom="true" />
```

</RelativeLayout>

Following is the content of the res/values/string.xml.

```
<resources>
    <string name="app_name">My Application</string>
</resources>
```

Following is the content of AndroidManifest.xml file.

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

```

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.web.baou.myapplication" >
  <uses-permission android:name="android.permission.INTERNET" />
  <application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
    <activity
      android:name=".MainActivity"
      android:label="@string/app_name" >
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
  </application>
</manifest>

```

Let's try to run your WebView application. To run the app from Android studio, open one of your project's activity files and click Run Android Studio Run Icon icon from the toolbar. Android studio will display as shown below

Now just specify a url on the url field and press the browse button that appears, to launch the website. But before that please make sure that you are connected to the internet. After pressing the button, the following screen would appear

Note. By just changing the url in the url field, your WebView will open your desired website.

Here is an example demonstrating the use of WIFI. It creates a basic application that open your wifi and close your wifi

To experiment with this example, you need to run this on an actual device on which wifi is turned on. follow the steps:

1. You will use Android studio to create an Android application under a package com.wifi.baou.myapplication.
2. Modify src/MainActivity.java file to add WebView code.
3. Modify the res/layout/activity_main to add respective XML components
4. Modify the AndroidManifest.xml to add the necessary permissions
5. Run the application and choose a running android device and install the application on it and verify the results.

Following is the content of the modified main activity file src/MainActivity.java.

```
package com.wifi.baou.myapplication;
import android.net.wifi.WifiManager;
import android.os.Bundle;
import android.app.Activity;
import android.content.Context;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class MainActivity extends Activity {
    Button enableButton,disableButton;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        enableButton=(Button)findViewById(R.id.button1);
        disableButton=(Button)findViewById(R.id.button2);

        enableButton.setOnClickListener(new OnClickListener(){
            public void onClick(View v){
```

```

        WifiManager wifi = (WifiManager)
getSystemService(Context.WIFI_SERVICE);
        wifi.setWifiEnabled(true);
    }
});

```

```

        disableButton.setOnClickListener(new OnClickListener(){
            public void onClick(View v){
                WifiManager wifi = (WifiManager)
getSystemService(Context.WIFI_SERVICE);
                wifi.setWifiEnabled(false);
            }
        });
    }
}

```

Following is the modified content of the xml res/layout/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:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context=".MainActivity">

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/imageView"
        android:src="@drawable/abc"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true" />

    <Button
        android:id="@+id/button1"

```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="76dp"
        android:text="Enable Wifi"
        android:layout_centerVertical="true"
        android:layout_alignEnd="@+id/imageView" />

<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Disable Wifi"
    android:layout_marginBottom="93dp"
    android:layout_alignParentBottom="true"
    android:layout_alignStart="@+id/imageView" />
</RelativeLayout>

```

Following is the content of AndroidManifest.xml file.

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.wifi.baou.myapplication" >
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

```

```
</intent-filter>
</activity>
</application>
</manifest>
```

Let's try to run your application. I assume you have connected your actual Android Mobile device with your computer.

- To run the app from Android studio,
- open one of your project's activity files and click Run android studio Run Icon icon from the toolbar.
- Before starting your application,
- Android studio will display following window to select an option where you want to run your Android application.

Telephony API **13**

Let's see the simple example of TelephonyManager that prints information of the telephony services.

activity_main.xml

Drag one textview from the palette, now the xml file will look like this.

```
<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:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginLeft="38dp"
        android:layout_marginTop="30dp"
        android:text="Phone Details:" />

</RelativeLayout>
```

MainActivity.java

```
package com.baou.telephonymanager;
import android.os.Bundle;
import android.app.Activity;
import android.content.Context;
import android.telephony.TelephonyManager;
import android.view.Menu;
import android.widget.TextView;

public class MainActivity extends Activity {
    TextView textView1;
```

```

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

    textView1=(TextView)findViewById(R.id.textView1);

    //Get the instance of TelephonyManager
    TelephonyManager
tm=(TelephonyManager) getSystemService(Context.TELEPHONY_SERVICE);

    //Calling the methods of TelephonyManager the returns the information
    String IMEINumber=tm.getDeviceId();
    String subscriberID=tm.getDeviceId();
    String SIMSerialNumber=tm.getSimSerialNumber();
    String networkCountryISO=tm.getNetworkCountryIso();
    String SIMCountryISO=tm.getSimCountryIso();
    String softwareVersion=tm.getDeviceSoftwareVersion();
    String voiceMailNumber=tm.getVoiceMailNumber();

    //Get the phone type
    String strphoneType="";

    int phoneType=tm.getPhoneType();

    switch (phoneType)
    {
        case (TelephonyManager.PHONE_TYPE_CDMA):
            strphoneType="CDMA";
            break;
        case (TelephonyManager.PHONE_TYPE_GSM):
            strphoneType="GSM";
            break;
        case (TelephonyManager.PHONE_TYPE_NONE):
            strphoneType="NONE";
            break;
    }

    //getting information if phone is in roaming

```



```

boolean isRoaming=tm.isNetworkRoaming();

String info="Phone Details:\n";
info+="\n IMEI Number:"+IMEINumber;
info+="\n SubscriberID:"+subscriberID;
info+="\n Sim Serial Number:"+SIMSerialNumber;
info+="\n Network Country ISO:"+networkCountryISO;
info+="\n SIM Country ISO:"+SIMCountryISO;
info+="\n Software Version:"+softwareVersion;
info+="\n Voice Mail Number:"+voiceMailNumber;
info+="\n Phone Network Type:"+strphoneType;
info+="\n In Roaming? :"+isRoaming;
textView1.setText(info);//displaying the information in the textView
}
}

```

AndroidManifest.xml

Need to write READ_PHONE_STATE permission in the AndroidManifest.xml file.

File: AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:androclass="http://schemas.android.com/apk/res/android"
    package="com.javatpoint.telephonymanager"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity

```

```
        android:name="com.javatpoint.telephonymanager.MainActivity"
        android:label="@string/app_name" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>
</manifest>
```

Run same code in emulator or device and check your telephony API detail on screen

Create an activity_main.xml file in layout folder containing ScrollView and ListView.

File: 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:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.test.searchview.MainActivity">
    <ListView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/lv1"
        android:divider="#ad5"
        android:dividerHeight="2dp"
        android:layout_below="@+id/searchView"/>
    <SearchView
        android:id="@+id/searchView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:queryHint="Search Here"
        android:iconifiedByDefault="false"
        android:layout_alignParentTop="true"
    />
</RelativeLayout>
```

File: MainActivity.java

```

package com.baou.pgdmad.searchview;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.AdapterView;
import android.widget.Filter;
import android.widget.ListView;
import android.widget.SearchView;
import android.widget.Toast;
import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {
    SearchView searchView;
    ListView listView;
    ArrayList<String> list;
    ArrayAdapter<String > adapter;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        searchView = (SearchView) findViewById(R.id.searchView);
        listView = (ListView) findViewById(R.id.lv1);

        list = new ArrayList<>();
        list.add("Apple");
        list.add("Banana");
        list.add("Pineapple");
        list.add("Orange");
        list.add("Lychee");
        list.add("Gavava");
        list.add("Peech");
        list.add("Melon");
        list.add("Watermelon");
        list.add("Papaya");

        adapter = new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_1,list);

```

```

listView.setAdapter(adapter);
searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
    @Override
    public boolean onQueryTextSubmit(String query) {
        if(list.contains(query)){
            adapter.getFilter().filter(query);
        }else{
            Toast.makeText(MainActivity.this, "No Match
found",Toast.LENGTH_LONG).show();
        }
        return false;
    }
    @Override
    public boolean onQueryTextChange(String newText) {
        // adapter.getFilter().filter(newText);
        return false;
    }
});
}
}

```

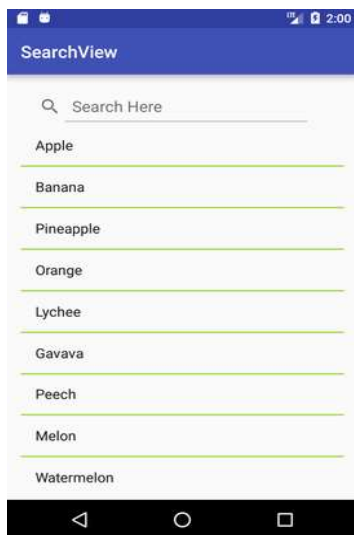


Figure-14 Item List

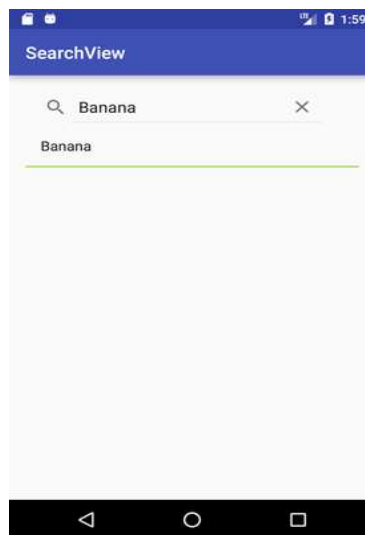


Figure-25 Search View

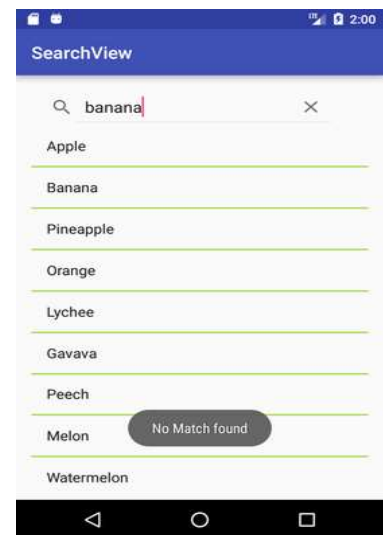


Figure-26 Search item notification

Create Android Application (Step & Description):

1. You will use Android studio IDE to create an Android application and name it as MyLocationApp under a package com.baou.mylocation.apps.
2. add src/GPSTracker.java file and add required code.
3. Modify src/MainActivity.java file and add required code as shown below to take care of getting current location and its equivalent address.
4. Modify layout XML file res/layout/activity_main.xml to add all GUI components which include three buttons and two text views to show location/address.
5. Modify res/values/strings.xml to define required constant values
6. Modify AndroidManifest.xml as shown below
7. Run the application to launch Android emulator and verify the result of the changes done in the application.

Code for main activity file MainActivity.java.

```
package com.baou.mylocation.apps;
import android.Manifest;
import android.app.Activity;
import android.os.Bundle;
import android.support.v4.app.ActivityCompat;
import android.test.mock.MockPackageManager;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends Activity {

    Button btnShowLocation;
    private static final int REQUEST_CODE_PERMISSION = 2;
    String mPermission = Manifest.permission.ACCESS_FINE_LOCATION;

    // GPSTracker class
```

```

GPSTracker gps;

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

    try {
        if (ActivityCompat.checkSelfPermission(this, mPermission)
            != MockPackageManager.PERMISSION_GRANTED) {

            ActivityCompat.requestPermissions(this, new String[]{mPermission},
                REQUEST_CODE_PERMISSION);

            // If any permission above not allowed by user, this condition will
            // execute every time, else your else part will work
        }
    } catch (Exception e) {
        e.printStackTrace();
    }

    btnShowLocation = (Button) findViewById(R.id.button);

    // show location button click event
    btnShowLocation.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View arg0) {
            // create class object
            gps = new GPSTracker(MainActivity.this);

            // check if GPS enabled
            if(gps.canGetLocation()){

                double latitude = gps.getLatitude();
                double longitude = gps.getLongitude();

                // \n is for new line
                Toast.makeText(getApplicationContext(), "Your Location is - \nLat: "

```

```

        + latitude + "\nLong: " + longitude, Toast.LENGTH_LONG).show();
    }
    else{
        // can't get location
        // GPS or Network is not enabled
        // Ask user to enable GPS/network in settings
        gps.showSettingsAlert();
    }
}
});
}
}

```

Code for main activity file GPSTracker.java.

```

package com.baou.mylocation.apps;
import android.app.AlertDialog;
import android.app.Service;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.os.IBinder;
import android.provider.Settings;
import android.util.Log;

public class GPSTracker extends Service implements LocationListener {

    private final Context mContext;

    // flag for GPS status
    boolean isGPSEnabled = false;

    // flag for network status

```



```

boolean isNetworkEnabled = false;

// flag for GPS status
boolean canGetLocation = false;

Location location; // location
double latitude; // latitude
double longitude; // longitude

// The minimum distance to change Updates in meters
private static final long MIN_DISTANCE_CHANGE_FOR_UPDATES = 10; // 10
meters

// The minimum time between updates in milliseconds
private static final long MIN_TIME_BW_UPDATES = 1000 * 60 * 1; // 1 minute

// Declaring a Location Manager
protected LocationManager locationManager;

public GPSTracker(Context context) {
    this.mContext = context;
    getLocation();
}

public Location getLocation() {
    try {
        locationManager = (LocationManager)
mContext.getSystemService(LOCATION_SERVICE);

        // getting GPS status
        isGPSEnabled =
locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);

        // getting network status
        isNetworkEnabled = locationManager
.isProviderEnabled(LocationManager.NETWORK_PROVIDER);

        if (!isGPSEnabled && !isNetworkEnabled) {
            // no network provider is enabled
        } else {

```

```

this.canGetLocation = true;
// First get location from Network Provider
if (isNetworkEnabled) {
    locationManager.requestLocationUpdates(
        LocationManager.NETWORK_PROVIDER,
        MIN_TIME_BW_UPDATES,
        MIN_DISTANCE_CHANGE_FOR_UPDATES, this);

    Log.d("Network", "Network");
    if (locationManager != null) {
        location = locationManager
            .getLastKnownLocation(LocationManager.NETWORK_PROVIDER);

        if (location != null) {
            latitude = location.getLatitude();
            longitude = location.getLongitude();
        }
    }
}

// if GPS Enabled get lat/long using GPS Services
if (isGPSEnabled) {
    if (location == null) {
        locationManager.requestLocationUpdates(
            LocationManager.GPS_PROVIDER,
            MIN_TIME_BW_UPDATES,
            MIN_DISTANCE_CHANGE_FOR_UPDATES, this);

        Log.d("GPS Enabled", "GPS Enabled");
        if (locationManager != null) {
            location = locationManager
                .getLastKnownLocation(LocationManager.GPS_PROVIDER);

            if (location != null) {
                latitude = location.getLatitude();
            }
        }
    }
}

```

```

        longitude = location.getLongitude();
    }
}
}
}

} catch (Exception e) {
    e.printStackTrace();
}
return location;
}
/**
 * Stop using GPS listener
 * Calling this function will stop using GPS in your app
 */

public void stopUsingGPS(){
    if(locationManager != null){
        locationManager.removeUpdates(GPSTracker.this);
    }
}
/**
 * Function to get latitude
 */
public double getLatitude(){
    if(location != null){
        latitude = location.getLatitude();
    }
    // return latitude
    return latitude;
}
/**
 * Function to get longitude
 */

```

```

public double getLongitude(){
    if(location != null){
        longitude = location.getLongitude();
    }

    // return longitude
    return longitude;
}

/**
 * Function to check GPS/wifi enabled
 * @return boolean
 * */

public boolean canGetLocation() {
    return this.canGetLocation;
}

/**
 * Function to show settings alert dialog
 * On pressing Settings button will launch Settings Options
 * */

public void showSettingsAlert(){
    AlertDialog.Builder alertDialog = new AlertDialog.Builder(mContext);

    // Setting Dialog Title
    alertDialog.setTitle("GPS is settings");

    // Setting Dialog Message
    alertDialog.setMessage("GPS is not enabled. Do you want to go to settings
menu?");

    // On pressing Settings button
    alertDialog.setPositiveButton("Settings", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog,int which) {
            Intent intent = new
Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
            mContext.startActivity(intent);

```

```

    }
});

// on pressing cancel button
alertDialog.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int which) {
        dialog.cancel();
    }
});

// Showing Alert Message
    alertDialog.show();
}

@Override
public void onLocationChanged(Location location) {
}

@Override
public void onProviderDisabled(String provider) {
}

@Override
public void onProviderEnabled(String provider) {
}

@Override
public void onStatusChanged(String provider, int status, Bundle extras) {
}

@Override
public IBinder onBind(Intent arg0) {
    return null;
}
}

```

Code of res/layout/activity_main.xml file
 <?xml version = "1.0" encoding = "utf-8"?>

```

<LinearLayout xmlns:android = "http://schemas.android.com/apk/res/android"
    android:layout_width = "fill_parent"
    android:layout_height = "fill_parent"
    android:orientation = "vertical" >

```

```

<Button
    android:id = "@+id/button"
    android:layout_width = "fill_parent"
    android:layout_height = "wrap_content"
    android:text = "getlocation"/>

```

```

</LinearLayout>

```

Code of res/values/strings.xml to define two new constants –

```

<?xml version = "1.0" encoding = "utf-8"?>
<resources>
    <string name = "app_name">MyLocationApp </string>
</resources>

```

Code of AndroidManifest.xml –

```

<?xml version = "1.0" encoding = "utf-8"?>
<manifest xmlns:android = "http://schemas.android.com/apk/res/android"
    package = "com.baou.mylocation.apps">
    <uses-permission android:name = "android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name = "android.permission.INTERNET" />
    <application
        android:allowBackup = "true"
        android:icon = "@mipmap/ic_launcher"
        android:label = "@string/app_name"
        android:supportsRtl = "true"
        android:theme = "@style/AppTheme">
        <activity android:name = ".MainActivity">
            <intent-filter>
                <action android:name = "android.intent.action.MAIN" />
                <category android:name = "android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>

```

```
</intent-filter>
</activity>
</application>

</manifest>
```

Let's try to run your MyLocation application. I assume that, you have connected your actual Android Mobile device with your computer. To run the app from Android Studio.



Figure-30 Ready for collect geo location information

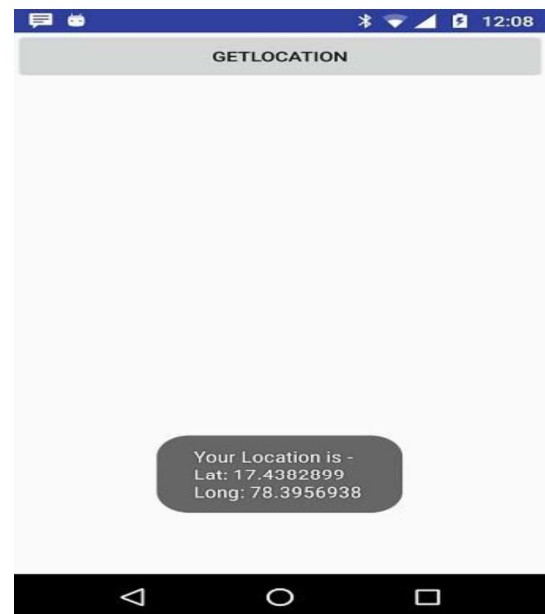


Figure-31 Received geolocation information in toast message

Bluetooth 16

This program relies on the `BluetoothAdapter` class to control the device's Bluetooth adapter, the `BluetoothDevice` class to represent the state of the connected device, and the `BluetoothSocket` class to represent sockets for listening for and making connections:

```
package com.baou.bluetooth;
import android.os.Handler;
import android.os.Message;
public class BtHelperHandler extends Handler {
    public enum MessageType {
        STATE,
        READ,
        WRITE,
        DEVICE,
        NOTIFY;
    }
    public Message obtainMessage(MessageType message, int count, Object obj) {
        return obtainMessage(message.ordinal(), count, -1, obj);
    }
    public MessageType getMessageType(int ordinal) {
        return MessageType.values()[ordinal];
    }
}
```

The `BtHelperHandler` class defines some constants and provides a little bit of wrapper code that makes message-related methods cleaner. `BtSPPHelper.java` is what encapsulates our use of the Bluetooth Serial Port Protocol (SPP):

```
package com.baou.bluetooth;
import java.io.IOException;
import java.io.InputStream;
```



```

import java.io.OutputStream;
import java.util.UUID;
import com.baou.R;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothServerSocket;
import android.bluetooth.BluetoothSocket;
import android.content.Context;
import android.os.Bundle;
import android.os.Message;
import android.util.Log;
/**
 * Helper class that runs AsyncTask objects for communicating with a Bluetooth
 * device. This code is derived from the BluetoothChat example, but modified in
 * several ways to increase modularity and generality: The Handler is in a
 * separate class to make it easier to drop into other components.
 *
 * Currently this only does Bluetooth SPP. This can be generalized to other
 * services.
 */
public class BtSPPHelper {
    // Debugging
    private final String TAG = getClass().getSimpleName();
    private static final boolean D = true;
    public enum State {
        NONE,
        LISTEN,
        CONNECTING,
        CONNECTED;
    }
}
// Name for the SDP record when creating server socket
private static final String NAME = "BluetoothTest";
// Unique UUID for this application
private static final UUID SPP_UUID =
UUID.fromString("00001101-0000-1000-8000-00805F9B34FB");

```

```

// Member fields
private final BluetoothAdapter mAdapter;
private final BtHelperHandler mHandler;
private AcceptThread mAcceptThread;
private ConnectThread mConnectThread;
private ConnectedThread mConnectedThread;
private State mState;
private Context mContext;

/**
 * Constructor. Prepares a new Bluetooth SPP session.
 * @param context The UI Activity Context
 * @param handler A Handler to send messages back to the UI Activity
 */
public BtSPPHelper(Context context, BtHelperHandler handler) {
mContext = context;
mAdapter = BluetoothAdapter.getDefaultAdapter();
mState = State.NONE;
mHandler = handler;
}

/**
 * Set the current state of the chat connection
 * @param state The current connection state
 */
private synchronized void setState(State state) {

if (D) Log.d(TAG, "setState() " + mState + " -> " + state);
mState = state;
// Give the new state to the Handler so the UI Activity can update
mHandler.obtainMessage(BtHelperHandler.MessageType.STATE,
-1, state).sendToTarget();
}

/**
 * Return the current connection state.
 */
public synchronized State getState() {

```

```

return mState;
}

/**
 * Start the session. Start AcceptThread to begin a
 * session in listening (server) mode.
 *
 * Typically, call this in onResume()
 */
public synchronized void start() {
    if (D) Log.d(TAG, "start");
    // Cancel any thread attempting to make a connection
    if (mConnectThread != null) {mConnectThread.cancel(); mConnectThread =
    null;}
    // Cancel any thread currently running a connection
    if (mConnectedThread != null) {
        mConnectedThread.cancel();
        mConnectedThread = null;
    }
    // Start the thread to listen on a BluetoothServerSocket
    if (mAcceptThread == null) {
        mAcceptThread = new AcceptThread();
        mAcceptThread.start();
    }
    setState(State.LISTEN);
}

/**
 * Start the ConnectThread to initiate a connection to a remote device.
 * @param device The BluetoothDevice to connect
 */
public synchronized void connect(BluetoothDevice device) {
    if (D) Log.d(TAG, "connect to: " + device);
    // Cancel any thread attempting to make a connection
    if (mState == State.CONNECTING) {
        if (mConnectThread != null) {
            mConnectThread.cancel();

```

```

        mConnectThread = null;
    }
}
// Cancel any thread currently running a connection
    if (mConnectedThread != null) {
        mConnectedThread.cancel();
        mConnectedThread = null;
    }
// Start the thread to connect with the given device
    mConnectThread = new ConnectThread(device);
    mConnectThread.start();
    setState(State.CONNECTING);
}
/**
 * Start the ConnectedThread to begin managing a Bluetooth connection
 *
 * @param socket
 * The BluetoothSocket on which the connection was made
 * @param device
 * The BluetoothDevice that has been connected
 */
private synchronized void connected(BluetoothSocket socket,
BluetoothDevice device) {
    if (D)
        Log.d(TAG, "connected");
        // Cancel the thread that completed the connection
    if (mConnectThread != null) {
        mConnectThread.cancel();
        mConnectThread = null;
    }
// Cancel any thread currently running a connection
    if (mConnectedThread != null) {
        mConnectedThread.cancel();
        mConnectedThread = null;
    }
}

```

```

// Cancel the accept thread because we only want to connect to one
// device
    if (mAcceptThread != null) {
        mAcceptThread.cancel();
        mAcceptThread = null;
    }
// Start the thread to manage the connection and perform transmissions
    mConnectedThread = new ConnectedThread(socket);
    mConnectedThread.start();
// Send the name of the connected device back to the UI Activity
    mHandler.obtainMessage(BtHelperHandler.MessageType.DEVICE, -1,
        device.getName()).sendToTarget();
    setState(State.CONNECTED);
}
/**
 * Stop all threads
 */
public synchronized void stop() {
    if (D) Log.d(TAG, "stop");
    if (mConnectThread != null) {
        mConnectThread.cancel();
        mConnectThread = null;
    }
    if (mConnectedThread != null) {
        mConnectedThread.cancel();
        mConnectedThread = null;
    }
    if (mAcceptThread != null) {
        mAcceptThread.cancel();
        mAcceptThread = null;
    }
    setState(State.NONE);
}
/**
 * Write to the ConnectedThread in an unsynchronized manner

```

```

* @param out The bytes to write
* @see ConnectedThread#write(byte[])
*/
public void write(byte[] out) {
    ConnectedThread r;
    // Synchronize a copy of the ConnectedThread
    synchronized (this) {
        if (mState != State.CONNECTED) return;
        r = mConnectedThread;
    }
    // Perform the write unsynchronized
    r.write(out);
}

private void sendErrorMessage(int messageld) {
    setState(State.LISTEN);
    mHandler.obtainMessage(BtHelperHandler.MessageType.NOTIFY, -1,
        mContext.getResources().getString(messageld)).sendToTarget();
}
/**
* This thread listens for incoming connections.
*/
private class AcceptThread extends Thread {
    // The local server socket
    private final BluetoothServerSocket mmServerSocket;
    public AcceptThread() {
        BluetoothServerSocket tmp = null;
        // Create a new listening server socket
        try {
            tmp = mAdapter.listenUsingRfcommWithServiceRecord(NAME, SPP_UUID);
        } catch (IOException e) {
            Log.e(TAG, "listen() failed", e);
        }
        mmServerSocket = tmp;
    }
    public void run() {

```

```

if (D) Log.d(TAG, "BEGIN mAcceptThread" + this);
setName("AcceptThread");
BluetoothSocket socket = null;
// Listen to the server socket if we're not connected
while (mState != BtSPPHelper.State.CONNECTED) {
try {
// This is a blocking call and will only return on a
// successful connection or an exception
socket = mmServerSocket.accept();
} catch (IOException e) {
Log.e(TAG, "accept() failed", e);
break;
}
// If a connection was accepted
if (socket != null) {
synchronized (BtSPPHelper.this) {
switch (mState) {
case LISTEN:
case CONNECTING:
// Situation normal. Start the connected thread.
connected(socket, socket.getRemoteDevice());
break;
case NONE:
case CONNECTED:
// Either not ready or already connected.
// Terminate new socket.
try {
socket.close();
} catch (IOException e) {
Log.e(TAG, "Could not close unwanted socket", e);
}
break;
}
}
}
}
}

```

```

}
if (D) Log.i(TAG, "END mAcceptThread");
}
public void cancel() {
if (D) Log.d(TAG, "cancel " + this);
try {
mmServerSocket.close();
} catch (IOException e) {
Log.e(TAG, "close() of server failed", e);
}
}
}
/**
 * This thread runs while attempting to make an outgoing connection
 * with a device. It runs straight through; the connection either
 * succeeds or fails.
 */
private class ConnectThread extends Thread {
private final BluetoothSocket mmSocket;
private final BluetoothDevice mmDevice;
public ConnectThread(BluetoothDevice device) {
mmDevice = device;
BluetoothSocket tmp = null;
// Get a BluetoothSocket for a connection with the
// given BluetoothDevice
try {
tmp = device.createRfcommSocketToServiceRecord(SPP_UUID);
} catch (IOException e) {
Log.e(TAG, "create() failed", e);
}
mmSocket = tmp;
}
public void run() {
Log.i(TAG, "BEGIN mConnectThread");
setName("ConnectThread");

```



```

// Always cancel discovery because it will slow down a connection
mAdapter.cancelDiscovery();
// Make a connection to the BluetoothSocket
try {
// This is a blocking call and will only return on a
// successful connection or an exception
mmSocket.connect();
} catch (IOException e) {
sendMessage(R.string.bt_unable);
// Close the socket
try {
mmSocket.close();
} catch (IOException e2) {
Log.e(TAG, "unable to close() socket during connection failure",
e2);
}
// Start the service over to restart listening mode
BtSPPHelper.this.start();
return;
}
// Reset the ConnectThread because we're done
synchronized (BtSPPHelper.this) {
mConnectThread = null;
}
// Start the connected thread
connected(mmSocket, mmDevice);
}
public void cancel() {
try {
mmSocket.close();
} catch (IOException e) {
Log.e(TAG, "close() of connect socket failed", e);
}
}
}
}

```

```

/**
 * This thread runs during a connection with a remote device.
 * It handles all incoming and outgoing transmissions.
 */
private class ConnectedThread extends Thread {
private final BluetoothSocket mmSocket;
private final InputStream mmInStream;
private final OutputStream mmOutStream;
public ConnectedThread(BluetoothSocket socket) {
Log.d(TAG, "create ConnectedThread");
mmSocket = socket;
InputStream tmpIn = null;
OutputStream tmpOut = null;
// Get the BluetoothSocket input and output streams
try {
tmpIn = socket.getInputStream();
tmpOut = socket.getOutputStream();
} catch (IOException e) {
Log.e(TAG, "temp sockets not created", e);
}
mmInStream = tmpIn;
mmOutStream = tmpOut;
}
public void run() {
Log.i(TAG, "BEGIN mConnectedThread");
byte[] buffer = new byte[1024];
int bytes;
// Keep listening to the InputStream while connected
while (true) {
try {
// Read from the InputStream
bytes = mmInStream.read(buffer);
// Send the obtained bytes to the UI Activity
mHandler.obtainMessage(BtHelperHandler.MessageType.READ,
bytes, buffer).sendToTarget();

```

```

} catch (IOException e) {
    Log.e(TAG, "disconnected", e);
    sendErrorMessage(R.string.bt_connection_lost);
    break;
}
}
}
/**
 * Write to the connected OutputStream.
 * @param buffer The bytes to write
 */
public void write(byte[] buffer) {
    try {
        mmOutputStream.write(buffer);
        // Share the sent message back to the UI Activity
        mHandler.obtainMessage(BtHelperHandler.MessageType.WRITE, -1, buffer)
            .sendToTarget();
    } catch (IOException e) {
        Log.e(TAG, "Exception during write", e);
    }
}
public void cancel() {
    try {
        mmSocket.close();
    } catch (IOException e) {
        Log.e(TAG, "close() of connect socket failed", e);
    }
}
}
}
}
}
}
}
}

```

The BtSPPHelper class brings the use of these classes together, and also contains the definition of private Thread subclasses that listen for, establish, and run connections. This is also where the java.io package meets Android Bluetooth: the Bluetooth Socket objects contain methods that return references to InputStream and

Output Stream objects to be used to read and write data on the socket connection:

```
package com.baou.bluetooth;

import java.util.Set;
import com.baou.finch.R;
import android.app.Activity;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.Window;
import android.view.View.OnClickListener;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.AdapterView.OnItemClickListener;
/**
 * Derived from the BluetoothChat example, an activity that enables
 * picking a paired or discovered Bluetooth device
 */
public class DeviceListActivity extends Activity {
    // Debugging
    private static final String TAG = "DeviceListActivity";
    private static final boolean D = true;
    // Return Intent extra
    public static String EXTRA_DEVICE_ADDRESS = "device_address";
    // Member fields
    private BluetoothAdapter mBtAdapter;
    private ArrayAdapter<String> mPairedDevicesArrayAdapter;
```

```

private ArrayAdapter<String> mNewDevicesArrayAdapter;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    // Set up the window
    setContentView(R.layout.device_list);
    // Set result CANCELED in case the user backs out
    setResult(Activity.RESULT_CANCELED);
    // Initialize the button to perform device discovery
    Button scanButton = (Button) findViewById(R.id.button_scan);
    scanButton.setOnClickListener(new OnClickListener() {
        public void onClick(View v) {
            doDiscovery();
            v.setVisibility(View.GONE);
        }
    });
    // Initialize array adapters. One for already paired devices and
    // one for newly discovered devices
    mPairedDevicesArrayAdapter = new ArrayAdapter<String>(this,
    R.layout.device_name);
    mNewDevicesArrayAdapter = new ArrayAdapter<String>(this,
    R.layout.device_name);
    // Find and set up the ListView for paired devices
    ListView pairedListView = (ListView) findViewById(R.id.paired_devices);
    pairedListView.setAdapter(mPairedDevicesArrayAdapter);
    pairedListView.setOnItemClickListener(mDeviceClickListener);
    // Find and set up the ListView for newly discovered devices
    ListView newDevicesListView = (ListView) findViewById(R.id.new_devices);
    newDevicesListView.setAdapter(mNewDevicesArrayAdapter);
    newDevicesListView.setOnItemClickListener(mDeviceClickListener);
    // Register for broadcasts when a device is discovered
    IntentFilter filter = new IntentFilter(BluetoothDevice.ACTION_FOUND);
    this.registerReceiver(mReceiver, filter);
    // Register for broadcasts when discovery has finished
    filter = new IntentFilter(BluetoothAdapter.ACTION_DISCOVERY_FINISHED);

```

```

this.registerReceiver(mReceiver, filter);
// Get the local Bluetooth adapter
mBtAdapter = BluetoothAdapter.getDefaultAdapter();
// Get a set of currently paired devices
Set<BluetoothDevice> pairedDevices = mBtAdapter.getBondedDevices();
// If there are paired devices, add each one to the ArrayAdapter
if (pairedDevices.size() > 0) {
    findViewById(R.id.title_paired_devices).setVisibility(View.VISIBLE);
    for (BluetoothDevice device : pairedDevices) {
        mPairedDevicesArrayAdapter.add(device.getName() +
            "\n" + device.getAddress());
    }
} else {
    String noDevices =
        getResources().getText(R.string.none_paired).toString();
    mPairedDevicesArrayAdapter.add(noDevices);
}
}
@Override
protected void onDestroy() {
    super.onDestroy();
    // Make sure we're not doing discovery anymore
    if (mBtAdapter != null) {
        mBtAdapter.cancelDiscovery();
    }
    // Unregister broadcast listeners
    this.unregisterReceiver(mReceiver);
}
/**
 * Start device discovery with the BluetoothAdapter
 */
private void doDiscovery() {
    if (D) Log.d(TAG, "doDiscovery()");
    // Indicate scanning in the title
    setProgressBarIndeterminateVisibility(true);

```

```

setTitle(R.string.scanning);
// Turn on sub-title for new devices
findViewById(R.id.title_new_devices).setVisibility(View.VISIBLE);
// If we're already discovering, stop it
if (mBtAdapter.isDiscovering()) {
    mBtAdapter.cancelDiscovery();
}
// Request discovery from BluetoothAdapter
mBtAdapter.startDiscovery();
}
// The on-click listener for all devices in the ListViews
private OnItemClickListener mDeviceClickListener = new OnItemClickListener() {
    public void onItemClick(AdapterView<?> av, View v, int arg2, long arg3) {
        // Cancel discovery because it's costly and we're about to connect
        mBtAdapter.cancelDiscovery();
        // Get the device MAC address, which is the last 17 chars in the View
        String info = ((TextView) v).getText().toString();
        String address = info.substring(info.length() - 17);
        // Create the result Intent and include the MAC address
        Intent intent = new Intent();
        intent.putExtra(EXTRA_DEVICE_ADDRESS, address);
        // Set result and finish this Activity
        setResult(Activity.RESULT_OK, intent);
        finish();
    }
};
// The BroadcastReceiver that listens for discovered devices and
// changes the title when discovery is finished
private final BroadcastReceiver mReceiver = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        String action = intent.getAction();
        // When discovery finds a device
        if (BluetoothDevice.ACTION_FOUND.equals(action)) {
            // Get the BluetoothDevice object from the Intent

```

```

BluetoothDevice device =
intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
// If it's already paired, skip it, because it's been listed already
if (device.getBondState() != BluetoothDevice.BOND_BONDED) {
mNewDevicesArrayAdapter.add(
device.getName() + "\n" + device.getAddress());
}
// When discovery is finished, change the Activity title
} else if (BluetoothAdapter.ACTION_DISCOVERY_FINISHED.equals(action)) {
setProgressBarIndeterminateVisibility(false);
setTitle(R.string.select_device);
if (mNewDevicesArrayAdapter.getCount() == 0) {
String noDevices =
getResources().getText(R.string.none_found).toString();
mNewDevicesArrayAdapter.add(noDevices);
}
}
}
};

```

The DeviceListActivity class

This activity displays a dialog that lists known devices and enables the user to request a scan for devices. Unlike those parts of the app where Thread subclasses are used to implement asynchronous I/O and Handler subclasses pass the results to the UI thread, the startDiscovery method of the BluetoothAdapter class kicks off a separate thread and communicates results using broadcast intents. A BroadcastReceiver is used here to process those results.

The BtConsoleActivity class

The BtConsoleActivity class creates a chat-like activity for interacting with a Bluetooth device. The menus in this activity enable connecting to a device, and the main view in this activity is a scrolling list of data sent and received. At the bottom of the screen, there is an EditText view for entering text to be sent to the other end of the SPP connection.

Handler classes are used to glue the single-threaded UI to the threads that listen for, establish, and perform I/O on socket connections.

Connecting with Facebook

17

Log in With Facebook

Open up the main layout file `activity_main.xml` and add a Facebook login button below the `TextView`:

```
<com.facebook.login.widget.LoginButton
    android:id="@+id/facebookLoginButton"
    android:layout_width="wrap_content"
    android:layout_height="47dp"
    android:paddingBottom="15dp"
    android:paddingStart="10dp"
    android:paddingEnd="5dp"
    android:paddingTop="15dp"
    android:textSize="16sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.58" />
```

In `MainActivity`, create the following constants at the top of the class:

```
val EMAIL = "email"
val PUBLIC_PROFILE = "public_profile"
val USER_PERMISSION = "user_friends"
```

Inside the empty method `facebookSetup()`, add the following code:

```
callbackManager = CallbackManager.Factory.create()
facebookLoginButton.setOnClickListener {
    facebookLoginButton.setReadPermissions(Arrays.asList(EMAIL,
PUBLIC_PROFILE, USER_PERMISSION))
    facebookLoginButton.registerCallback(callbackManager,
object: FacebookCallback<LoginResult> {
    override fun onSuccess(loginResult: LoginResult) {
```

```

    }

    override fun onCancel() {
    }

    override fun onError(exception: FacebookException) {
        Toast.makeText(context,exception.localizedMessage,
        Toast.LENGTH_SHORT).show()
    }
})
}

```

This code first initializes the CallbackManager Facebook property that was declared but uninitialized in the starter project. It then adds a click listener for the Facebook login button. Inside the click listener, it provides the permissions needed to read the email, public profile and friends of the user. It also logs in the user by returning the AccessToken.

Then in onActivityResult(), pass the result onto the CallbackManager:

```
callbackManager.onActivityResult(requestCode, resultCode, data)
```

In the onSuccess of the callback, you'll get the user's profile by using Facebook's Graph API. You'll then send the user to the Share screen. First, we need to talk to the Graph API.

User Profile from the Graph API

Create an object called Helper in a new package com.baousocial.socialapps.util.

Once created, write the method getFacebookUserProfileWithGraphApi() inside of it:

```

object Helper {

    fun getFacebookUserProfileWithGraphApi(context: Context) {

        if (AccessToken.getCurrentAccessToken() != null){
            val activity = context as Activity
            val request = GraphRequest.newMeRequest(
                AccessToken.getCurrentAccessToken()

```

```

    ) { jsonObject, _ ->
        val email = jsonObject?.get("email")?.toString() ?: ""
        val name = jsonObject.get("name").toString()
        val profileObjectImage =
        jsonObject?.getJSONObject("picture")?.getJSONObject("data")?.get("url").toString()
    }

    val parameters = Bundle()
    parameters.putString("fields", "id,name,link,picture.type(large), email")
    request.parameters = parameters
    request.executeAsync()
}
}
}

```

This method uses a call to `GraphRequest.newMeRequest()` to fetch the `userid`, `name`, `picture` and `email` of the user who is currently logged in.

To keep things clean, create a package `com.baousocial.socialapps.model` and create a class in the package called `UserModel` to contain the user's data after the Graph API returns the results.

Your `UserModel` class would look something like this:

```

class UserModel(val name: String, val userName: String, val profilePictureUrl: String,
val socialNetwork: SocialNetwork) : Serializable

enum class SocialNetwork {
    Facebook, Twitter
}

```

Now we have to create the enum class `SocialNetwork` in the same class; you could create a separate file for that if you wish. The enum class is only for identifying which social network account the user is currently logged in with.

Head back to `Helper` where you'll now write the method that will help in sending the user to the `Share` screen.

```

fun startShareActivity(context: Context, user: UserModel) {

```

```

val activity = context as Activity
val intent = Intent(context, ShareActivity::class.java)
intent.putExtra("user", user)
activity.startActivity(intent)
activity.finish()
}

```

This code takes the passed-in UserModel and sends it to the ShareActivity.

Go back to the method getFacebookUserProfileWithGraphApi() and after the line:

```

val profileObjectImage =
jsonObject?.getJSONObject("picture")?.getJSONObject("data")?.get("url").toString()
?: ""

```

add the following:

```

val user = UserModel(name, email, profileObjectImage, SocialNetwork.Facebook)
startShareActivity(context, user)

```

These lines convert the user's info to a UserModel and pass it into the method startShareActivity().

After completing that, go back to MainActivity. In the onSuccess inside of facebookSetup(), write:

```

        Helper.getFacebookUserProfileWithGraphApi(context)

```

The user should only be sent to the Share screen when the user has a valid AccessToken, and this can happen only in the onSuccess block of code.

Additionally, you need to set up a few things in the ShareActivity.

Create a UserModel property in the class:

```

lateinit var user: UserModel

```

And inside onCreate(), add:

```
user = intent.extras.get("user") as UserModel
setData(user)
```

This piece of code is getting the passed in UserModel from the Intent method and passing the data to a new method setData().

The setData() method simply sets up the data in the UI, and includes conditionals that take slightly different actions depending on whether the logged in network is Facebook or Twitter.

```
fun setData(user: UserModel) {
    nameTextView.text = user.name
    userNameTextView.text =
        if (user.socialNetwork == SocialNetwork.Twitter) "@${user.userName}"
        else user.userName
    connectedWithTextView.text =
        if (user.socialNetwork == SocialNetwork.Twitter)
        "${connectedWithTextView.text} Twitter"
        else "${connectedWithTextView.text} Facebook"
    characterLimitTextView.visibility =
        if (user.socialNetwork == SocialNetwork.Twitter) View.VISIBLE
        else View.GONE
    postButton.text =
        if (user.socialNetwork == SocialNetwork.Twitter) "POST"
        else "CREATE POST"

    Picasso.with(this).load(user.profilePictureUri).placeholder(R.drawable.ic_user)
        .into(profileImageView)
    postEditText.visibility = View.GONE
}
```

Now, run your app then tap on Continue with Facebook. You'll be asked to give your app the permission to access the information.

Sharing on Facebook

It's time to move on to posting a status to Facebook. For this, you need to change a few things.

Facebook recently changed its documentation and has removed the permission that was once required for the using the Graph API to share something on a user's timeline. The alternative for that is now using the Facebook Share SDK.

Open the app build.gradle file, and add the following dependency in it:

```
implementation 'com.facebook.android:facebook-share:[4,5]'
```

Additionally, in your AndroidManifest.xml add the following line within the application tag:

```
<provider
    android:authorities="com.facebook.app.FacebookContentProvider{@string/facebook_app_id}"
    android:name="com.facebook.FacebookContentProvider"
    android:exported="true"/>
```

Now, open the ShareActivity class and write the method for posting status to Facebook:

```
fun postStatusToFacebook() {
    val builder = AlertDialog.Builder(this)
    builder.setTitle("Share Link")

    val input = EditText(this@ShareActivity)
    val lp = LinearLayout.LayoutParams(
        LinearLayout.LayoutParams.MATCH_PARENT,
        LinearLayout.LayoutParams.MATCH_PARENT)
    input.layoutParams = lp
    builder.setView(input)

    builder.setPositiveButton(android.R.string.ok) { dialog, p1 ->
        val link = input.text
        var isValid = true
        if (link.isBlank()) {
            isValid = false
        }
    }
}
```

```

    }

    if (isValid) {
        val content = ShareLinkContent.Builder()
            .setContentUrl(Uri.parse(link.toString()))
            .build()
        ShareDialog.show(this, content)
    }

    dialog.dismiss()
}

builder.setNegativeButton(android.R.string.cancel) { dialog, p1 ->
    dialog.cancel()
}

builder.show()
}

```

This code will present an alert dialog to allow the user to enter a link to share, and then show the user the Facebook share dialog. We're not doing any validation on the link other than to check that it's not blank; you'd want to do some validation to make sure it's a valid URL.

In later versions of the Facebook Share SDK, including the one you're using in Socialapps, you must provide some type of content to share. Your options are links, photos, videos, and other multimedia. See the Facebook Share SDK documentation for more details.

Next, in the `postButtonAction()` method, inside the `setOnClickListener`, add a call to the new function:

postStatusToFacebook()

Build and run the app again. You'll need to tap logout on the Facebook button and re-connect. In a production app, you'll want to save the logged in state of the user so that they don't have to log in again.

Click on CREATE POST. Now, try posting something to Facebook:

Logging Out of Facebook

Logging out is simply a one-line code, but, for logging out, you need to perform two additional tasks. You'll now write a method in your `ShareActivity` that'll do these tasks:

```
fun sendToMainActivity() {  
    LoginManager.getInstance().logout()  
    finish()  
    val intent = Intent(this, MainActivity::class.java)  
    startActivity(intent)  
}
```

Going over the above: the first line of code allows a user to log out of Facebook. The rest of the lines finish the current activity and take a user to `MainActivity`. Finally, call this method inside the `onOptionsItemSelected` like this:

```
R.id.action_logout -> {  
    sendToMainActivity()  
    return true  
}
```

Once you tap the Logout button on top-right of the Share screen, you'll be logged out from the app and taken to the Home screen.

Sensor Example:

```
SensorManager sManager;  
sManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
```

Listing sensors supported by device:

Example:

```
List<Sensor> dSensors = sManager.getSensorList(Sensor.TYPE_ALL);
```

Determine existence of specific type of sensor on a device:

getDefaultSensor() requires the type of sensor for checking existence as a parameter.

Example:

```
if (sManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD) != null){  
    // Success! There's a magnetometer.
```

```
    Toast.makeText(getApplicationContext(),"Its  
Magnetometer...",Toast.LENGTH_SHORT).show();
```

```
} else {
```

```
    // Failure! No magnetometer.
```

```
    Toast.makeText(getApplicationContext(),"Not available  
Magnetometer...",Toast.LENGTH_SHORT).show();
```

```
}
```

Monitoring Sensor Events**EXAMPLE:**

Use the onSensorChanged() method for monitoring light sensor data which display in TextView.

```

public class SensorActivity extends Activity implements SensorEventListener {
    private SensorManager sensorManager;
    private Sensor mLight;
    @Override

    public final void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.main);

        sensorManager = (SensorManager)
getSystemService(Context.SENSOR_SERVICE);
        mLight = sensorManager.getDefaultSensor(Sensor.TYPE_LIGHT);
    }
    @Override
    public final void onAccuracyChanged(Sensor sensor, int accuracy) {
        // Code to perform when sensor accuracy changed.
    }
    @Override
    public final void onSensorChanged(SensorEvent event) {
        // The light sensor returns a single value.
        // Many sensors return 3 values, one for each axis.
        float lux = event.values[0];
        // Do something with this sensor value.
    }
    @Override
    protected void onResume() {
        super.onResume();
        sensorManager.registerListener(this, mLight,
SensorManager.SENSOR_DELAY_NORMAL);
//The default delay is specified when the registerListener() method is invoked.
    }
    @Override
    protected void onPause() {
        super.onPause();
        sensorManager.unregisterListener(this);
    }
}

```

Runtime Sensor Detection

EXAMPLE:

```
private SensorManager sensorManager;
sensorManager = (SensorManager)
getSystemService(Context.SENSOR_SERVICE);
if (sensorManager.getDefaultSensor(Sensor.TYPE_PRESSURE) != null){
    // Success! There's a pressure sensor.
} else {
    // Failure! No pressure sensor.
}
```

Specific sensor configurations using Google Play filters

The Google Play Store targeted applications consume this feature. The `<uses-feature>` elements in your **manifest file** to filter your application from devices that do not have the appropriate sensor configuration for your application.

EXAMPLE:

```
<uses-feature android:name="android.hardware.sensor.accelerometer"
    android:required="true" />
```

EXAMPLE:

```
private SensorManager sensorManager;
private Sensor sensor;
sensorManager = (SensorManager)
getSystemService(Context.SENSOR_SERVICE);
sensor = sensorManager.getDefaultSensor(Sensor.TYPE_GRAVITY);
```

- **The Linear Accelerometer:** Generally this sensor used for gesture detection. It provides three-dimensional vector with acceleration along each device axis, excluding gravity.

EXAMPLE:

```
private SensorManager sensorManager;
```

```

private Sensor sensor;
sensorManager = (SensorManager)
getSystemService(Context.SENSOR_SERVICE);
sensor =
sensorManager.getDefaultSensor(Sensor.TYPE_LINEAR_ACCELERATION);

```

The sensor provides acceleration data by calculating following.

linear acceleration = acceleration - acceleration due to gravity

- **The Rotation Vector Sensor:** It characterizes the orientation of the device as a combination of an angle and an axis, in which the device has rotated through an angle θ around an axis (x, y, or z).

EXAMPLE:

```

private SensorManager sensorManager;
private Sensor sensor;

sensorManager = (SensorManager)
getSystemService(Context.SENSOR_SERVICE);
sensor = sensorManager.getDefaultSensor(Sensor.TYPE_ROTATION_VECTOR);

```

- **The Significant Motion Sensor:** This sensor invoked and disable it by them self. It invoked when significant motion is detected and then it disables itself. This sensors are might be lead to change user location so its generally used for walking, biking, or sitting in a moving car.

EXAMPLE:

```

private SensorManager sensorManager;
private Sensor sensor;
private TriggerEventListener triggerEventListener;

sensorManager = (SensorManager)
getSystemService(Context.SENSOR_SERVICE);

sensor =
sensorManager.getDefaultSensor(Sensor.TYPE_SIGNIFICANT_MOTION);
triggerEventListener = new TriggerEventListener() {
    @Override
    public void onTrigger(TriggerEvent event) {
        // Do work
    }
}

```

```
    }  
};
```

- **The Step Counter Sensor:** It provides the number of steps taken by the user since the last reboot while the sensor was activated. The step counter has more latency (up to 10 seconds) but more accuracy than the step detector sensor.

EXAMPLE:

```
private SensorManager sensorManager;  
private Sensor sensor;  
sensorManager = (SensorManager)  
getSystemService(Context.SENSOR_SERVICE);  
sensor = sensorManager.getDefaultSensor(Sensor.TYPE_STEP_COUNTER);
```

- **The Step Detector Sensor:** This sensor fire an event each time the user takes a step. The latency is expected to be below 2 seconds.

EXAMPLE:

```
private SensorManager sensorManager;  
private Sensor sensor;  
ssensorManager = (SensorManager)  
getSystemService(Context.SENSOR_SERVICE);  
sensor = sensorManager.getDefaultSensor(Sensor.TYPE_STEP_DETECTOR);
```

- **The Game Rotation Vector Sensor:** It is identical to the Rotation vector sensor just except it does not use the geomagnetic field. Therefore the Y axis does not point north but instead to some other reference. That reference is allowed to drift by the same order of magnitude as the gyroscope drifts around the Z axis.

EXAMPLE:

```
private SensorManager sensorManager;  
private Sensor sensor;  
sensorManager = (SensorManager)  
getSystemService(Context.SENSOR_SERVICE);  
sensor = sensorManager.getDefaultSensor(  
Sensor.TYPE_GAME_ROTATION_VECTOR);
```

- **The Geomagnetic Rotation Vector Sensor:** It is same as Rotation vector sensor, but it uses a magnetometer instead of a gyroscope. The accuracy of this sensor is lower than the normal rotation vector sensor, but the power consumption is reduced. Only use this sensor if you want to collect some rotation information in the background without draining too much battery. This sensor is most useful when used in conjunction with batching.

EXAMPLE:

```
private SensorManager sensorManager;
private Sensor sensor;
sensorManager = (SensorManager)
getSystemService(Context.SENSOR_SERVICE);
sensor =
sensorManager.getDefaultSensor(Sensor.TYPE_GEOMAGNETIC_ROTATION_
VECTOR);
```

- **The Geomagnetic Field Sensor:** This sensor monitor changes in the earth's magnetic field.

EXAMPLE:

```
private SensorManager sensorManager;
private Sensor sensor;
sensorManager = (SensorManager)
getSystemService(Context.SENSOR_SERVICE);
sensor = sensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);
```

- **The Uncalibrated Magnetometer:** It is same as the geomagnetic field sensor, but it never apply the hard iron calibration to the magnetic field. Factory calibration and temperature compensation are still applied to the magnetic field. The uncalibrated magnetometer is useful to handle bad hard iron estimations.

EXAMPLE:

```
private SensorManager sensorManager;
private Sensor sensor;

sensorManager = (SensorManager)
getSystemService(Context.SENSOR_SERVICE);
```

```
sensor =  
sensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD_UNCALIB  
RATED);
```

- **The Proximity Sensor:** The proximity sensor returns two value near or far the object from device.

EXAMPLE:

```
private SensorManager sensorManager;  
private Sensor sensor;  
  
sensorManager = (SensorManager)  
getSystemService(Context.SENSOR_SERVICE);  
sensor = sensorManager.getDefaultSensor(Sensor.TYPE_PROXIMITY);
```

- **Device Orientation Calculation**

There are three types of orientation is available: Azimuth (degrees of rotation about the -z axis), Pitch (degrees of rotation about the x axis) and Roll (degrees of rotation about the y axis).

EXAMPLE:

```
private SensorManager sensorManager;  
  
// Rotation matrix based on current readings from accelerometer and  
magnetometer.  
final float[] rotationMatrix = new float[9];  
SensorManager.getRotationMatrix(rotationMatrix, null,  
    accelerometerReading, magnetometerReading);  
// Express the updated rotation matrix as three orientation angles.  
final float[] orientationAngles = new float[3];  
SensorManager.getOrientation(rotationMatrix, orientationAngles);
```

EXAMPLE:

```
public class SensorActivity extends Activity implements SensorEventListener {  
private SensorManager sensorManager;  
private Sensor pressure;  
  
@Override  
public final void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);
```

```

        setContentView(R.layout.main);

        // Get an instance of the sensor service, and use that to get an instance of
        // a particular sensor.
        sensorManager = (SensorManager)
getSystemService(Context.SENSOR_SERVICE);
        pressure = sensorManager.getDefaultSensor(Sensor.TYPE_PRESSURE);
    }
    @Override
    public final void onAccuracyChanged(Sensor sensor, int accuracy) {
        // Do something here if sensor accuracy changes.
    }
    @Override
    public final void onSensorChanged(SensorEvent event) {
        float millibarsOfPressure = event.values[0];
        // Do something with this sensor data.
    }
    @Override
    protected void onResume() {
        // Register a listener for the sensor.
        super.onResume();
        sensorManager.registerListener(this, pressure,
SensorManager.SENSOR_DELAY_NORMAL);
    }
    @Override
    protected void onPause() {
        // Be sure to unregister the sensor when the activity pauses.
        super.onPause();
        sensorManager.unregisterListener(this);
    }
}

```


Let us discuss the SAMPLE CODE for `NfcAdapter.CreateNdefMessageCallback`

```
package com.example.android.beam;
import android.app.Activity;
import android.content.Intent;
import android.nfc.NdefMessage;
import android.nfc.NdefRecord;
import android.nfc.NfcAdapter;
import android.nfc.NfcAdapter.CreateNdefMessageCallback;
import android.nfc.NfcEvent;
import android.os.Bundle;
import android.os.Parcelable;
import android.widget.TextView;
import android.widget.Toast;
import java.nio.charset.Charset;
public class Beam extends Activity implements CreateNdefMessageCallback {
    NfcAdapter nfcAdapter;
    TextView textView;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        TextView textView = (TextView) findViewById(R.id.textView);
        // Check for available NFC Adapter
        nfcAdapter = NfcAdapter.getDefaultAdapter(this);
        if (nfcAdapter == null) {
            Toast.makeText(this, "NFC is not available", Toast.LENGTH_LONG).show();
            finish();
            return;
        }
        // Register callback
        nfcAdapter.setNdefPushMessageCallback(this, this);
    }
}
```

```

}
@Override
public NdefMessage createNdefMessage(NfcEvent event) {
    String text = ("Beam me up, Android!\n\n" +
        "Beam Time: " + System.currentTimeMillis());
    NdefMessage msg = new NdefMessage(
        new NdefRecord[] { createMime(
            "application/vnd.com.example.android.beam", text.getBytes())
        /**
         * The Android Application Record (AAR) is commented out. When a device
         * receives a push with an AAR in it, the application specified in the AAR
         * is guaranteed to run. The AAR overrides the tag dispatch system.
         * You can add it back in to guarantee that this
         * activity starts when receiving a beamed message. For now, this code
         * uses the tag dispatch system.
         */
        //,NdefRecord.createApplicationRecord("com.example.android.beam")
    });
    return msg;
}
@Override
public void onResume() {
    super.onResume();
    // Check to see that the Activity started due to an Android Beam
    if (NfcAdapter.ACTION_NDEF_DISCOVERED.equals(getIntent().getAction())) {
        processIntent(getIntent());
    }
}
@Override
public void onNewIntent(Intent intent) {
    // onResume gets called after this to handle the intent
    setIntent(intent);
}

/**

```

```

    * Parses the NDEF Message from the intent and prints to the TextView
    */
    void processIntent(Intent intent) {
        textView = (TextView) findViewById(R.id.textView);
        Parcelable[] rawMsgs = intent.getParcelableArrayExtra(
            NfcAdapter.EXTRA_NDEF_MESSAGES);
        // only one message sent during the beam
        NdefMessage msg = (NdefMessage) rawMsgs[0];
        // record 0 contains the MIME type, record 1 is the AAR, if present
        textView.setText(new String(msg.getRecords()[0].getPayload()));
    }
}

```

Note that this code comments out an AAR, which you can remove. If you enable the AAR, the application specified in the AAR always receives the Android Beam message. If the application is not present, Google Play is started to download the application. Therefore, the following intent filter is not technically necessary for Android 4.0 devices or later if the AAR is used:

```

<intent-filter>
    <action android:name="android.nfc.action.NDEF_DISCOVERED"/>
    <category android:name="android.intent.category.DEFAULT"/>
    <data android:mimeType="application/vnd.com.example.android.beam"/>
</intent-filter>

```

Speech and Gestures Recognizer

20

EXAMPLE:

Create simple Speech to Text.

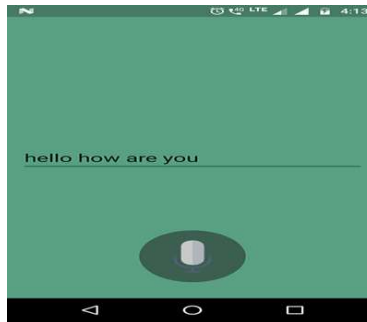


Figure-54 Layout for Speech to Text

Create ImageAsset icon_record named in drawable folder or download it

Right click on Project folder generally name app->New->ImageAsset->icon_record->Choose Clipart for record icon->select Circular shape->Next->Finish

Add two color in res->values->colors.xml

```
<color name="colorButton">#3d5f51</color>
<color name="colorButtonActive">#1bea95</color>
```

Change the style to NoActionBar as well in res->values->styles.xml

```
<resources>
<style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">
  <!-- Customize your theme here. -->
  <item name="colorPrimary">@color/colorPrimary</item>
  <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
  <item name="colorAccent">@color/colorAccent</item>
</style>
</resources>
```

Create round record icon by creating round_default.xml inside the drawable folder

```
<?xml version="1.0" encoding="utf-8"?>
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
  <item>
    <shape android:shape="oval">
      <solid android:color="@color/colorButton" />
    </shape>
  </item>
  <item
    android:bottom="20dp"
    android:drawable="@drawable/icon_record"
    android:left="20dp"
    android:right="20dp"
    android:top="20dp" />
</layer-list>
```

Create one more file named round_active.xml inside the drawable folder

```
<?xml version="1.0" encoding="utf-8"?>
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
  <item>
    <shape android:shape="oval">
      <solid android:color="@color/colorButtonActive" />
    </shape>
  </item>
  <item
    android:bottom="20dp"
    android:drawable="@drawable/icon_record"
    android:left="20dp"
    android:right="20dp"
    android:top="20dp" />
</layer-list>
```

Create button_background.xml for button background setting inside drawable folder

```

<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:drawable="@drawable/round_default"
android:state_pressed="false" />
  <item android:drawable="@drawable/round_active" android:state_pressed="true"
/>
</selector>

```

Now go for actual UI design under activity_main.xml for implementation read comments inside code

```

<?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:background="@color/colorPrimary"
  android:padding="12dp"
  tools:context="com.example.speechtotext.MainActivity">
  <EditText
    android:id="@+id/editText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_centerVertical="true"
    android:hint="Targeted Text Output Here From Speech"
    android:textAppearance="@style/Base.TextAppearance.AppCompat.Large" />
  <ImageButton
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="20dp"
    android:background="@drawable/button_background" />
</RelativeLayout>

```

Add RECORD_AUDIO permission in the AndroidManifest.xml

```
<uses-permission android:name="android.permission.RECORD_AUDIO" />
```

Now open MainActivity.java for actual code

```
package com.example.speechtotext;
```

```
import android.Manifest;
```

```
import android.content.Intent;
```

```
import android.content.pm.PackageManager;
```

```
import android.net.Uri;
```

```
import android.os.Build;
```

```
import android.provider.Settings;
```

```
import android.speech.RecognitionListener;
```

```
import android.speech.RecognizerIntent;
```

```
import android.speech.SpeechRecognizer;
```

```
import android.support.v4.content.ContextCompat;
```

```
import android.support.v7.app.AppCompatActivity;
```

```
import android.os.Bundle;
```

```
import android.view.MotionEvent;
```

```
import android.view.View;
```

```
import android.widget.EditText;
```

```
import java.util.ArrayList;
```

```
import java.util.Locale;
```

```
public class MainActivity extends AppCompatActivity {  
    final Button recordbtn=(Button) findViewById(R.id.button);
```

```
    @Override
```

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

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
        /*Create function checkPermission() outside of onCreate() to check the device is  
        running android marshmallow or ahead*/
```

```
        checkPermission();
```

```
        final EditText editText = (EditText)findViewById(R.id.editText);
```

```

        final SpeechRecognizer mSpeechRecognizer =
SpeechRecognizer.createSpeechRecognizer(this);
        final Intent mSpeechRecognizerIntent = new
Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);

mSpeechRecognizerIntent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL
,
        RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
mSpeechRecognizerIntent.putExtra(RecognizerIntent.EXTRA_LANGUAGE,
        Locale.getDefault());

/*Create RecognitionListener from Speech Recognizer object for interpret the text
from recording and display in EditText*/
mSpeechRecognizer.setRecognitionListener(new RecognitionListener() {
    @Override
    public void onReadyForSpeech(Bundle bundle) {
        }
    @Override
    public void onBeginningOfSpeech() {
        }
    @Override
    public void onRmsChanged(float v) {
        }
    @Override
    public void onBufferReceived(byte[] bytes) {
        }
    @Override
    public void onEndOfSpeech() {
        }
    @Override
    public void onError(int i) {
        }
    @Override
    public void onResults(Bundle bundle) {
        //getting all the matches
        ArrayList<String> matches = bundle
                .getStringArrayList(SpeechRecognizer.RESULTS_RECOGNITION);
    }
}

```



```

        //displaying the first match
        if (matches != null)
            editText.setText(matches.get(0));
    }
    @Override
    public void onPartialResults(Bundle bundle) {
    }
    @Override
    public void onEvent(int i, Bundle bundle) {
    }
});

/*Create onTouchListener of Button object to get the speech and start listening, and
after removing the finger from the button, it will stop listening.*/
recordbtn.setOnTouchListener(new View.OnTouchListener() {
    @Override
    public boolean onTouch(View view, MotionEvent motionEvent) {
        switch (motionEvent.getAction()) {
            case MotionEvent.ACTION_UP:
                mSpeechRecognizer.stopListening();
                editText.setHint("Targeted Text From Speech");
                break;

            case MotionEvent.ACTION_DOWN:
                mSpeechRecognizer.startListening(mSpeechRecognizerIntent);
                editText.setText("");
                editText.setHint("Listening...");
                break;
        }
        return false;
    }
});

}

/* Outside of onCreate define said previously method.*/
private void checkPermission() {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {

```

```

        if (!(ContextCompat.checkSelfPermission(this,
Manifest.permission.RECORD_AUDIO) ==
PackageManager.PERMISSION_GRANTED)) {
            Intent intent = new
Intent(Settings.ACTION_APPLICATION_DETAILS_SETTINGS,
            Uri.parse("package:" + getPackageName()));
            startActivity(intent);
            finish();
        }
    }
}
}
}

```

GESTURE RECOGNIZER

EXAMPLE:

Create touch event for Activity

For this override the `onTouchEvent()` Method with `MotionEvent`. Use the `getActionMasked()` to extract the action which is user performed from the event parameter.

```

public class MainActivity extends Activity {
    ...
    // This example shows an Activity, but you would use the same approach if
    // you were subclassing a View.
    @Override
    public boolean onTouchEvent(MotionEvent event){
        int action = MotionEventCompat.getActionMasked(event);
        switch(action) {
            case (MotionEvent.ACTION_DOWN) :
                Log.d(DEBUG_TAG,"Action was DOWN");
                return true;
            case (MotionEvent.ACTION_MOVE) :
                Log.d(DEBUG_TAG,"Action was MOVE");
                return true;
            case (MotionEvent.ACTION_UP) :

```

```

        Log.d(DEBUG_TAG,"Action was UP");
        return true;
    case (MotionEvent.ACTION_CANCEL) :
        Log.d(DEBUG_TAG,"Action was CANCEL");
        return true;
    case (MotionEvent.ACTION_OUTSIDE) :
        Log.d(DEBUG_TAG,"Movement occurred outside bounds " +
            "of current screen element");
        return true;
    default :
        return super.onTouchEvent(event);
    }
}

```

Create touch events for any single view

For this create any view which object we want to refer with `setOnTouchListener()` method.

```

View myView = findViewById(R.id.my_view);
myView.setOnTouchListener(new OnTouchListener() {
    public boolean onTouch(View v, MotionEvent event) {
        // ... Respond to touch events
        return true;
    }
});

```

Detect Common Gestures

Android provides the `GestureDetector` class for detecting common gestures. Some of the gestures it supports include `onDown()`, `onLongPress()`, `onFling()`, and so on. You can use `GestureDetector` in conjunction with the `onTouchEvent()` method described above.

```

public class MainActivity extends Activity implements
    GestureDetector.OnGestureListener,
    GestureDetector.OnDoubleTapListener {
    private static final String DEBUG_TAG = "Gestures";
    private GestureDetectorCompat mDetector;
    // Called when the activity is first created.
    @Override

```

```

public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    // Instantiate the gesture detector with the
    // application context and an implementation of
    // GestureDetector.OnGestureListener
    mDetector = new GestureDetectorCompat(this, this);
    // Set the gesture detector as the double tap
    // listener.
    mDetector.setOnDoubleTapListener(this);
}
@Override
public boolean onTouchEvent(MotionEvent event){
    if (this.mDetector.onTouchEvent(event)) {
        return true;
    }
    return super.onTouchEvent(event);
}
@Override
public boolean onKeyDown(MotionEvent event) {
    Log.d(DEBUG_TAG, "onDown: " + event.toString());
    return true;
}
@Override
public boolean onFling(MotionEvent event1, MotionEvent event2,
    float velocityX, float velocityY) {
    Log.d(DEBUG_TAG, "onFling: " + event1.toString() + event2.toString());
    return true;
}
@Override
public void onLongPress(MotionEvent event) { Log.d(DEBUG_TAG,
"onLongPress: " + event.toString());
}
@Override
public boolean onScroll(MotionEvent event1, MotionEvent event2, float distanceX,

```

```

        float distanceY) {
    Log.d(DEBUG_TAG, "onScroll: " + event1.toString() + event2.toString());
    return true;
}
@Override
public void onShowPress(MotionEvent event) {
    Log.d(DEBUG_TAG, "onShowPress: " + event.toString());
}
@Override
public boolean onSingleTapUp(MotionEvent event) {
    Log.d(DEBUG_TAG, "onSingleTapUp: " + event.toString());
    return true;
}
@Override
public boolean onDoubleTap(MotionEvent event) {
    Log.d(DEBUG_TAG, "onDoubleTap: " + event.toString());
    return true;
}
@Override
public boolean onDoubleTapEvent(MotionEvent event) {
    Log.d(DEBUG_TAG, "onDoubleTapEvent: " + event.toString());
    return true;
}
@Override
public boolean onSingleTapConfirmed(MotionEvent event) {
    Log.d(DEBUG_TAG, "onSingleTapConfirmed: " + event.toString());
    return true;
}
}
}

```

યુનિવર્સિટી ગીત

સ્વાધ્યાય: પરમં તપ:

સ્વાધ્યાય: પરમં તપ:

સ્વાધ્યાય: પરમં તપ:

શિક્ષણ, સંસ્કૃતિ, સદ્ભાવ, દિવ્યબોધનું ધામ
ડૉ. બાબાસાહેબ આંબેડકર ઓપન યુનિવર્સિટી નામ;
સૌને સૌની પાંખ મળે, ને સૌને સૌનું આભ,
દશે દિશામાં સ્મિત વહે હો દશે દિશે શુભ-લાભ.

અભણ રહી અજ્ઞાનના શાને, અંધકારને પીવો ?
કહે બુદ્ધ આંબેડકર કહે, તું થા તારો દીવો;
શારદીય અજવાળા પહોંચ્યાં ગુર્જર ગામે ગામ
ધ્રુવ તારકની જેમ ઝળહળે એકલવ્યની શાન.

સરસ્વતીના મયૂર તમારે ફળિયે આવી ગહેકે
અંધકારને હડસેલીને ઉજાસના ફૂલ મહેકે;
બંધન નહીં કો સ્થાન સમયના જવું ન ઘરથી દૂર
ઘર આવી મા હરે શારદા દૈન્ય તિમિરના પૂર.

સંસ્કારોની સુગંધ મહેકે, મન મંદિરને ધામે
સુખની ટપાલ પહોંચે સૌને પોતાને સરનામે;
સમાજ કેરે દરિયે હાંકી શિક્ષણ કેરું વહાણ,
આવો કરીયે આપણ સૌ
ભવ્ય રાષ્ટ્ર નિર્માણ...
દિવ્ય રાષ્ટ્ર નિર્માણ...
ભવ્ય રાષ્ટ્ર નિર્માણ

