

## Assignment 6

- Title: Media player in Android
- Problem Statement:  
Design a mobile app for media player to load data from internal/external storage.
- Objective:
  - To understand design & development in Android Studio
  - To learn memory management in android.
- Outcome:
  - Design & implement music player application on android studio.
  - load songs from internal & external storage.
- Software & Hardware Packages:
  - Java JDK 8
  - android studio
  - 64 bit OS, intel i5+
  - 8GB RAM
- Theory:  
Music player requires
  - 1) List of songs
  - 2) Search Songs
  - 3) Song utilities screen - Play, Pause, etc.



### - Music Player (classes)

- 1) Content resolver - retrieve tracks
- 2) Media player - play audio
- 3) Media controller - control playback

### - Media Store:

It is a collection of Audio, Video & Images in MIME type.

collection of all mime type audio is under audio/\*

### - Media player (class)

Uses built in media player services  
To use it we have to call a static method create()

Syntax:

```
MediaPlayer mediaPlayer = MediaPlayer.create(
    this,
    R.raw.song
)
```

2) mediaPlayer.start()

3) mediaPlayer.pause()

4) mediaPlayer.reset()

### Other methods:

1) isPlaying()

2) SeekTo(position)

3) getCurrentPosition()

4) getDuration()



5) `reset()`

6) `release()` - releases resources attached.

7) `SetDataSource (File Descriptor id)`  
Selects tracks on given index

- Service:

Performs long running operations in the background.

A user defined service can be created by creating a class extending "service"

Some important methods of Android Services:

1) `onStartCommand()`

is called when a component requests to start a service

2) `onBind()`

is invoked when ever an application component calls the `bindService()` method to bind itself to a service.

3) `onUnbind()`:

is invoked when all clients get disconnected.

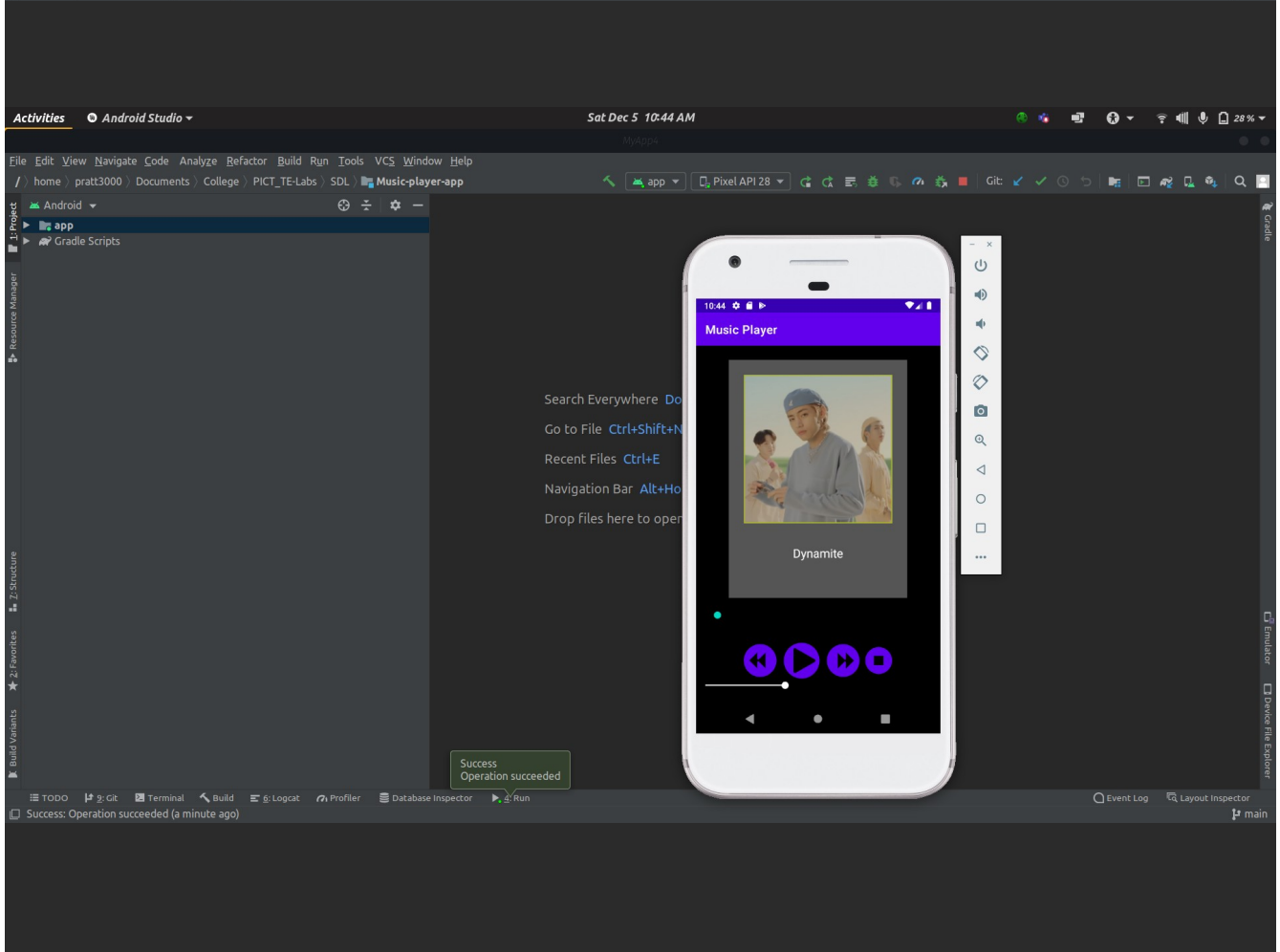
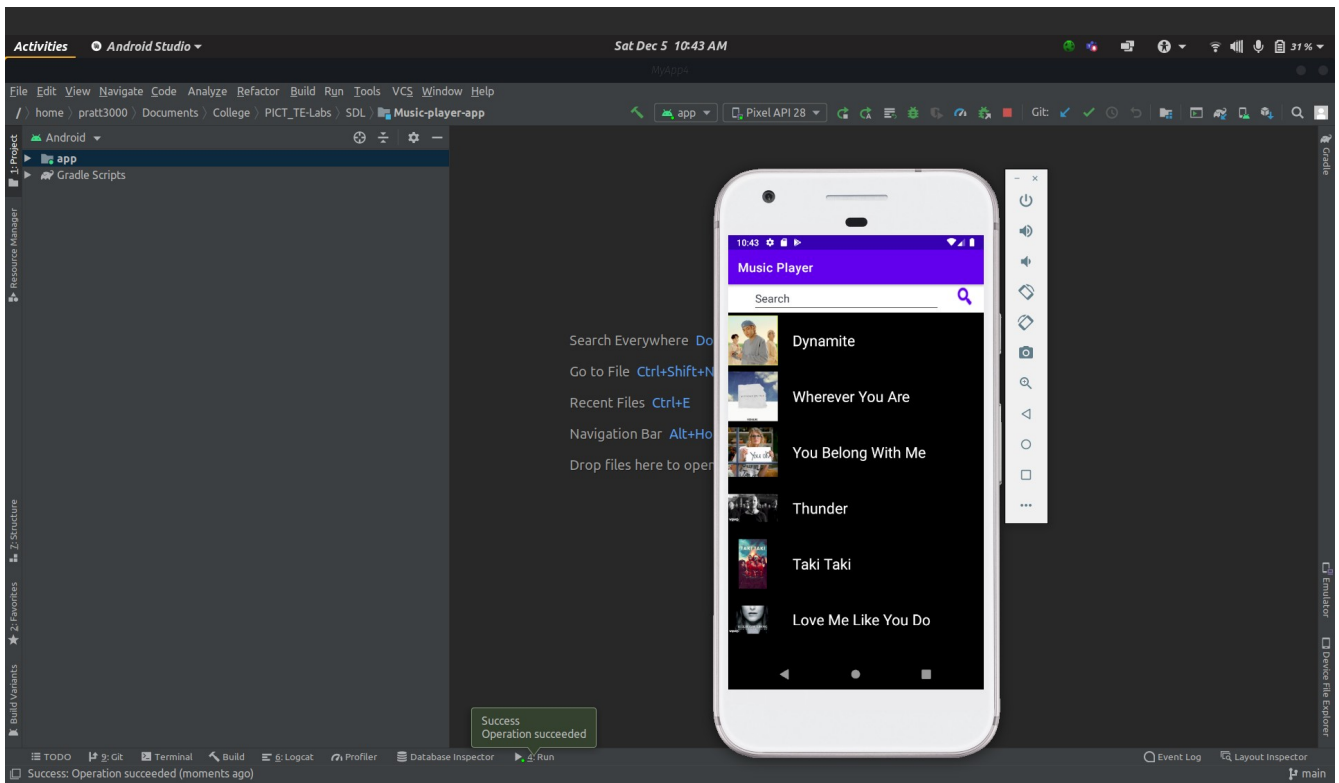
## • Test cases

Description	Expected output	result
1) asking permission for storage access	Lists audio file when granted else closes app	Success
2) list all songs	all files with MIME type audio/*	Success
3) Playing audio	audio is played when clicked on song title	Success
4) Audio control	Next, previous, pause work properly	Success

## • Conclusion:

In this assignment we implemented music player application in Android Studio.





```

package com.example.myapp4;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.Context;
import android.content.Intent;
import android.media.AudioManager;
import android.media.MediaPlayer;
import android.net.Uri;
import android.os.Bundle;
import android.os.Handler;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.SeekBar;
import android.widget.TextView;
import android.widget.Toast;

import java.io.IOException;
import java.util.ArrayList;
import java.util.Timer;
import java.util.TimerTask;

public class MainActivity extends AppCompatActivity {

    EditText name;
    Button button;
    static ArrayList<Song> data;

    private RecyclerView mRecyclerView;
    private SongsAdapter mAdapter;

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

        name = findViewById(R.id.dest);
        button = findViewById(R.id.load);

        RecyclerView songsList = findViewById(R.id.songsList);
        songsList.setLayoutManager(new LinearLayoutManager(this));
        data = new ArrayList<Song>();
        data.add(new Song("Dynamite", R.drawable.dynamite, R.raw.dynamite));
        data.add(new Song("Wherever You Are", R.drawable.kodaline_wherever_you_are,
R.raw.kodaline_wherever_you_are));
        data.add(new Song("You Belong With Me", R.drawable.you_belong_with_me,
R.raw.you_belong_with_me));
        data.add(new Song("Thunder", R.drawable.thunder, R.raw.thunder));
        data.add(new Song("Taki Taki", R.drawable.taki_taki, R.raw.taki_taki));
        data.add(new Song("Love Me Like You Do", R.drawable.love_me_like_you_do,
R.raw.love_me_like_you_do));
        data.add(new Song("Let Me Love You", R.drawable.let_me_love_you,
R.raw.let_me_love_you));
    }
}

```

```

        data.add(new Song("Lai Lai", R.drawable.lai_lai, R.raw.lai_lai));
        data.add(new Song("Galway Girl", R.drawable.galway_girl, R.raw.galway_girl));
        data.add(new Song("Faded", R.drawable.faded, R.raw.faded));
        data.add(new Song("Despacito", R.drawable.despacito_ft_daddy_yankee,
R.raw.despacito_ft_daddy_yankee));
        data.add(new Song("Believer", R.drawable.believer, R.raw.believer));
        data.add(new Song("Alone", R.drawable.alone, R.raw.alone));

        songsList.setAdapter(new SongsAdapter(getApplicationContext(), data));

        buildRecyclerView();

        name.addTextChangedListener(new TextWatcher() {
            @Override
            public void beforeTextChanged(CharSequence s, int start, int count, int after) {
            }
            @Override
            public void onTextChanged(CharSequence s, int start, int before, int count) {
            }
            @Override
            public void afterTextChanged(Editable s) {
                filter(s.toString());
            }
        });

    }

    private void filter(String text) {
        ArrayList<Song> filteredList = new ArrayList<>();
        for (Song item : data) {
            if (item.getTitle().toLowerCase().contains(text.toLowerCase())) {
                filteredList.add(item);
            }
        }
        mAdapter.filterList(filteredList);
    }

    private void buildRecyclerView() {
        mRecyclerView = findViewById(R.id.songsList);
        mRecyclerView.setHasFixedSize(true);
        layoutManager = new LinearLayoutManager(this);
        mAdapter = new SongsAdapter(this, data);
        mRecyclerView.setLayoutManager(layoutManager);
        mRecyclerView.setAdapter(mAdapter);
    }

```

```
}  
  
}
```

```
package com.example.myapp4;  
  
public class Song {  
    private String title;  
    private int imgId;  
    private int audioId;  
  
    public Song(String title, int imgId, int audioId) {  
        this.title = title;  
        this.imgId = imgId;  
        this.audioId = audioId;  
    }  
  
    public String getTitle() {  
        return title;  
    }  
  
    public void setTitle(String title) {  
        this.title = title;  
    }  
  
    public int getImgId() {  
        return imgId;  
    }  
  
    public void setImgId(int imgId) {  
        this.imgId = imgId;  
    }  
  
    public int getAudioId() {  
        return audioId;  
    }  
  
    public void setAudioId(int audioId) {  
        this.audioId = audioId;  
    }  
}
```

```
package com.example.myapp4;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Context;  
import android.content.Intent;  
import android.media.AudioManager;  
import android.media.MediaPlayer;  
import android.os.Bundle;  
import android.os.Handler;  
import android.util.Log;  
import android.view.View;  
import android.widget.Button;
```



```

import android.widget.ImageView;
import android.widget.SeekBar;
import android.widget.TextView;
import android.widget.Toast;

import com.bumptech.glide.Glide;

import java.io.IOException;
import java.util.ArrayList;

public class SongActivity extends AppCompatActivity {

    MediaPlayer player;
    Button btnplay, btnpause, btnstop, btnnext, btnback;
    SeekBar seekbar;
    Handler seekHandler = new Handler();
    private SeekBar seekBar;
    private AudioManager audioManager;
    ArrayList<Song> playlist;
    int i = 0;

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

        btnplay = (Button) findViewById(R.id.playbtn);
        btnpause = (Button) findViewById(R.id.pausebtn);
        btnstop = (Button) findViewById(R.id.stopbtn);
        btnnext = (Button) findViewById(R.id.nextbtn);
        btnback = (Button) findViewById(R.id.backbtn);
        seekbar = (SeekBar) findViewById(R.id.seekBar);

        // seekBar.setMax(audioManager.getStreamMaxVolume(AudioManager.STREAM_MUSIC));
        // seekBar.setProgress(audioManager.getStreamVolume(AudioManager.STREAM_MUSIC));
        seekBar = findViewById(R.id.seekBarVolume);
        // audioManager = (AudioManager)
        getApplicationContext().getSystemService(Context.AUDIO_SERVICE);

        player.setLooping(true);
        player.seekTo(0);

        player.setVolume(0.5f, 0.5f);
        seekBar.setProgress(50);
        seekBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
            @Override
            public void onProgressChanged(SeekBar seekBar, int progress, boolean isFromUser)
        {
            float volume = progress / 100f;
            player.setVolume(volume, volume);
        }

        @Override
        public void onStartTrackingTouch(SeekBar seekBar) {

```

```

    }

    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {

    }
});

btnplay.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        btnplay.setVisibility(View.GONE);
        btnpause.setVisibility(View.VISIBLE);
        player.start();
        seekbar.setMax(player.getDuration()/1000);
    }
});

btnpause.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        player.pause();
    }
});

btnstop.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        btnplay.setVisibility(View.VISIBLE);
        btnpause.setVisibility(View.GONE);
        if(player.isPlaying())
            player.stop();
        player = MediaPlayer.create(SongActivity.this,
playlist.get(i).getAudioId());

    }
});

btnnext.setOnClickListener(new View.OnClickListener(){
    @Override
    public void onClick(View v){
        player.stop();
        if (i == 12){
            i=0;
        }
        else{
            i++;
        }
        player = MediaPlayer.create(SongActivity.this,playlist.get(i).getAudioId());
        setMeta();
    }
});

```

```

        btnback.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View v){
                player.stop();
                if (i == 0){
                    i=12;
                }
                else{
                    i--;
                }
                player = MediaPlayer.create(SongActivity.this,playlist.get(i).getAudioId());
                setMeta();
            }
        });

        SongActivity.this.runOnUiThread(new Runnable() {

            @Override
            public void run() {
                if(player != null){
                    int mCurrentPosition = player.getCurrentPosition() / 1000;
                    seekbar.setProgress(mCurrentPosition);
                }
                seekHandler.postDelayed(this, 1000);
            }
        });

        seekbar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
            @Override
            public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
                if(player != null && fromUser){
                    player.seekTo(progress * 1000);
                }
            }

            @Override
            public void onStartTrackingTouch(SeekBar seekBar) {

            }

            @Override
            public void onStopTrackingTouch(SeekBar seekBar) {

            }
        });
    }

    @Override
    public void onBackPressed() {
        super.onBackPressed();
        if(player != null) {
            if(player.isPlaying()) {
                player.stop();
                player.release();
                player = null;
            }
        }
    }

    private void getIntentMethod() {

```



```

        i = getIntent().getIntExtra("position", 0);
        playlist = MainActivity.data;

        if(player != null)
            player.release();
        player = MediaPlayer.create(this, playlist.get(i).getAudioId());
        setMeta();
    }
    private void setMeta() {
        ImageView imgSong = findViewById(R.id.song_img);
        imgSong.setVisibility(View.VISIBLE);
        TextView txtSong = findViewById(R.id.song_name);

        Glide.with(this)
            .load(playlist.get(i).getImgId())
            .into(imgSong);

        txtSong.setText(playlist.get(i).getTitle());
    }
}

```

```

package com.example.myapp4;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;

import java.util.ArrayList;

public class SongsAdapter extends RecyclerView.Adapter<SongsAdapter.SongViewHolder> {

    private ArrayList<Song> data;
    private Context mContext;
    public SongsAdapter(Context mContext, ArrayList<Song> data) {
        this.mContext = mContext;
        this.data = data;
    }

    @NonNull
    @Override
    public SongViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        LayoutInflater inflater = LayoutInflater.from(parent.getContext());
        View view = inflater.inflate(R.layout.list_item_layout, parent, false);
        return new SongViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull SongViewHolder holder, int position) {
        Song currSong = data.get(position);
    }
}

```

```

        holder.txtTitle.setText(currSong.getTitle());
        Glide.with(mContext)
            .load(currSong.getImgId())
            .into(holder.imgIcon);

        holder.itemView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(mContext, SongActivity.class);
                intent.putExtra("position", position);
                intent.addFlags(intent.FLAG_ACTIVITY_NEW_TASK); //to avoid app crash
                mContext.startActivity(intent);
            }
        });
    }

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

    public class SongViewHolder extends RecyclerView.ViewHolder {

        ImageView imgIcon;
        TextView txtTitle;

        public SongViewHolder(@NonNull View itemView) {
            super(itemView);
            imgIcon = itemView.findViewById(R.id.imgIcon);
            txtTitle = itemView.findViewById(R.id.txtTitle);
        }
    }

    public void filterList(ArrayList<Song> filteredList) {
        data = filteredList;
        notifyDataSetChanged();
    }
}

```

```

<?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=".MainActivity">

    <EditText
        android:id="@+id/dest"
        android:layout_width="match_parent"
        android:layout_marginRight="70dp"
        android:backgroundTint="#000000"
        android:textColorHint="#353b48"
        android:layout_height="wrap_content"
        android:layout_marginLeft="40dp"
        android:hint="Search" />

```

```

<androidx.recyclerview.widget.RecyclerView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/songsList"
    android:background="@color/purple_200"
    android:layout_below="@+id/dest">

</androidx.recyclerview.widget.RecyclerView>

<Button
    android:id="@+id/load"
    android:layout_width="31dp"
    android:layout_height="39dp"
    android:layout_alignParentRight="true"
    android:layout_marginRight="15dp"
    android:background="@android:drawable/ic_search_category_default" />

</RelativeLayout>

```

```

<?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"
    android:gravity="center"
    android:background="#000000"
    tools:context=".SongActivity">

    <LinearLayout
        android:layout_width="300dp"
        android:layout_height="400dp"
        android:gravity="center"
        android:background="#55FFFFFF"
        android:orientation="vertical">

        <ImageView
            android:id="@+id/song_img"
            android:layout_width="250dp"
            android:layout_height="250dp"
            android:gravity="center"
            android:background="@drawable/mp3"
            android:scaleType="centerCrop"
            android:adjustViewBounds="true"
            android:visibility="invisible"
            android:alpha=".60">
        </ImageView>

        <TextView
            android:id="@+id/song_name"
            android:layout_width="wrap_content"
            android:layout_height="100dp"
            android:gravity="center"
            android:textColor="@color/white"
            android:textSize="20dp"
            android:text="Song Title">

```



```
</TextView>
```

```
</LinearLayout>
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:orientation="vertical">
```

```
    <SeekBar
```

```
        android:id="@+id/seekBar"  
        android:layout_width="320dp"  
        android:layout_height="wrap_content"  
        android:layout_marginStart="20dp"  
        android:layout_marginTop="20dp"  
        android:layout_marginEnd="10dp"  
        android:layout_marginBottom="20dp">
```

```
    </SeekBar>
```

```
    <LinearLayout
```

```
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:orientation="horizontal"  
        android:gravity="center"  
        android:layout_marginTop="15dp">
```

```
        <Button
```

```
            android:id="@+id/backbtn"  
            android:layout_width="55dp"  
            android:layout_height="55dp"  
            android:onClick="back"  
            android:layout_marginRight="10dp"  
            android:background="@drawable/back2" />
```

```
        <Button
```

```
            android:id="@+id/pausebtn"  
            android:layout_width="55dp"  
            android:layout_height="55dp"  
            android:onClick="pause"  
            android:layout_marginRight="10dp"  
            android:visibility="gone"  
            android:background="@drawable/pause" />
```

```
        <Button
```

```
            android:id="@+id/playbtn"  
            android:layout_width="65dp"  
            android:layout_height="65dp"  
            android:onClick="play"  
            android:layout_marginRight="10dp"  
            android:background="@drawable/play" />
```

```
        <Button
```

```
            android:id="@+id/nextbtn"  
            android:layout_width="55dp"
```

```

        android:layout_height="55dp"
        android:onClick="next"
        android:layout_marginRight="10dp"
        android:background="@drawable/next2" />

        <Button
            android:id="@+id/stopbtn"
            android:layout_width="45dp"
            android:layout_height="45dp"
            android:onClick="stop"

            android:background="@drawable/stop"/>

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal">

        <!--
            <Button-->
            <!--
                android:id="@+id/low"-->
            <!--
                android:layout_width="40dp"-->
            <!--
                android:layout_height="50dp"-->
            <!--
                android:layout_marginLeft="10dp"-->
            <!--
                android:layout_gravity="center"-->
            <!--
                android:text="-"-->
            <!--
                android:textSize="30dp"-->

        <!--
            </Button-->

            <SeekBar
                android:id="@+id/seekBarVolume"
                android:layout_width="300dp"
                android:layout_height="wrap_content"
                android:backgroundTint="#fff"
                android:progressTint="#fff"
                android:thumbTint="#fff" />

        <!--
            <Button-->
            <!--
                android:id="@+id/high"-->
            <!--
                android:layout_width="40dp"-->
            <!--
                android:layout_height="50dp"-->
            <!--
                android:text="+"-->
            <!--
                android:textSize="30dp"-->

        <!--
            </Button-->

        </LinearLayout>
    </LinearLayout>

</LinearLayout>

```

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    android:layout_width="match_parent"

```

```
android:layout_height="90dp"
android:orientation="horizontal"
android:background="@color/black"
xmlns:android="http://schemas.android.com/apk/res/android">
```

```
<ImageView
```

```
    android:id="@+id/imgIcon"
    android:layout_width="80dp"
    android:layout_height="80dp"
    android:src="@mipmap/ic_launcher"
    android:layout_gravity="center"
    android:scaleType="centerInside"
    android:adjustViewBounds="true"/>
```

```
<TextView
```

```
    android:id="@+id/textTitle"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/app_name"
    android:layout_gravity="center"
    android:textSize="24sp"
    android:textColor="@color/white"
    android:paddingLeft="24dp"/>
```

```
</LinearLayout>
```