SHILLONG COLLEGE

BOYCE ROAD, LAITUMKHRAH
SHILLONG - 793003, MEGHALAYA

ESTD - 1956

A PROJECT REPORT

**SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
OF
THE DEGREE OF
BACHELOR OF COMPUTER SCIENCE AND APPLICATIONS**

Submitted By,

**Student Name :Kitborlang Lyngdoh Kynshi
Roll No : P1400020
Reg.No : 14531 of 2013 – 2014**

Under the Guidance of

Sir BANTEI MUKHIM

Department of Computer Science And Applications

NORTH EASTERN HILLS UNIVERSITY

CERTIFIED THAT THIS IS A BONAFIDE RECORD OF THE PROJECT

SENSOR MUSIC PLAYER

Submitted for the partial fulfillment for the award of degree of
Bachelor of Computer Applications

By

Student Name : KITBORLANG LYNGDOH
Roll No : P1400020
Reg.No : 14531 of 2013 – 2014

Project Guide	**Head Of Department**	**Examiner**
Sir Bantei Mukhim	**Mrs. AiomMitri**	

Viva Voce held on:____________

ACKNOWLEDGEMENT

First of all,I would like to thank God who has always been there for me and especially for guiding me throughout my software development process.Also,I take my opportunity to thank and show my gratitude to my Guide I.e Sir Bantei Mukhim who always give me a helping hand especially in some most critical situation of my project.He has been there for me when I need most and he plays a major role in the completion of my project.I would also like to thank Miss. Aiom Mitri,our Head of Department,along with her other colleagues,for her acceppting my project topic.It's also a great pleasure to have my firends around me every time who have been suggesting various enhancement to be done on my project as well as supporting me fixing various problems in some of my coding and designing part, so it's an honour to have them on my back and I am very much thankful to them.Also,I'm very grateful to my parents,who are a great supporting role for me especially financially and most importantly for supporting me.Lastly,I take my chance to thank everyone who has been helping in moulding me through these 3 years of my under graduate life to build a better my and being a part for defining my future.

CONTENTS

PROJECT NAME

Sensor Music Player.

OBJECTIVES

The main Objective of my project is to control music playback in android devices by using hand hovering gestures over the device proximity sensor.

FEATURES

- Simple User Interface which is user friendly.
- Sensor control playback without touching device.e.g
- **Singlewave**-playnext
- **Doublewave** -play previous.
- **Stillwave**-pause/play.
- **Button control**:pause,play,next,previous,seek forward and backward.
- Enable-disable sensor control.
- Updating songlist dynamically
- Shuffling songs.

EXISTING AND PROPOSED SYSTEM

This app has a huge overhaul in comparison to the existing apps all in the android world.

Existing app don't have a proper interface design

Sensor control is not precise in existing projects and I'm trying to make this app better

No control over the sensor for enabling and disabling

REQUIREMENTS

ANDROID VERSION: Starting with android 4.4 to android 6.0.

Ram: 256mb and above.

CPU: Any new chipset with a clock speed of 1ghz and above.

A phone should have a Proximity sensor.

SOURCE CODE

ACTIVITY_PLAYER_MAIN.XML

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    android:background="@drawable/bgmain"
    tools:context=".PlayerMain"
    android:textColor="#000000">
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text=""
        android:gravity="center"
        android:textStyle="bold"
        android:background="#000000"

android:textColor="@color/background_floating_material_light"
        android:id="@+id/song_play"
        />
    <ListView
        android:id="@+id/song_list"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="@drawable/bgmain">

    </ListView>

</LinearLayout>
```

PLAYERMAIN.JAVA

```
package com.example.hp.player;

import android.app.Activity;
import android.app.ListActivity;
import android.content.ComponentName;
import android.content.Context;
import android.content.Intent;
import android.content.ServiceConnection;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.os.IBinder;
import android.os.SystemClock;
import android.provider.MediaStore;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.support.v7.internal.widget.AdapterViewCompat;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import android.net.Uri;
import android.content.ContentResolver;
import android.database.Cursor;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ListView;
import android.widget.MediaController.MediaPlayerControl;
import android.widget.TextView;
import android.widget.Toast;
import java.util.Timer;
import java.util.TimerTask;

import com.example.hp.player.MusicService.MusicBinder;

public class PlayerMain extends Activity implements
MediaPlayerControl,SensorEventListener {
    private ArrayList<Song> songList;
    private ListView songView;
    private MusicService musicSrv;
    private Intent playIntent;

    public boolean musicBound=false;
    private MusicController controller;
    private boolean paused=false;
    private boolean playbackPaused=false;
    private long checkTimer=0;
    private long timer;
    private boolean isChecked=false;
    SensorManager sm;
    Sensor sensor;
    long checkTime;
    TextView textView;
    Timer t;
    int i=0;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_player_main);
         textView=(TextView)findViewById(R.id.song_play);
        songView=(ListView)findViewById(R.id.song_list);
songList=new ArrayList<Song>();
    getSongList();
    Collections.sort(songList, new Comparator<Song>() {
        public int compare(Song a, Song b) {
            return a.getTitle().compareTo(b.getTitle());
        }
    });

    SongAdapter songAdt=new SongAdapter(this,songList);
    songView.setAdapter(songAdt);

    setController();
    t=new Timer();


}
private ServiceConnection musicConnection=new ServiceConnection() {

    public void onServiceConnected(ComponentName name, IBinder
service) {
        MusicBinder binder=(MusicBinder)service;
        musicSrv=binder.getService();
        musicSrv.setList(songList);
        musicBound=true;

    }


    public void onServiceDisconnected(ComponentName name) {
        musicBound=false;
    }
};
private void setController(){
    controller=new MusicController(this);
    controller.setPrevNextListeners(new View.OnClickListener() {

        public void onClick(View v) {
            playNext();
        }
    }, new View.OnClickListener() {
        public void onClick(View v) {
            playPrev();
        }

    });
    controller.setMediaPlayer(this);
    controller.setAnchorView(findViewById(R.id.song_list));
    controller.setEnabled(true);

}
private void playNext(){

    musicSrv.playNext();
    try {
        textView.setText(musicSrv.getName());
    }catch(Exception e){}
    if(playbackPaused){
        try{
        setController();}catch (Exception
e){Toast.makeText(PlayerMain.this,""+e,Toast.LENGTH_LONG).show();}
        playbackPaused=false;
    }
    try {
        controller.show(0);
    }catch (Exception e){}
}
private void playPrev(){

    musicSrv.playPrev();
    textView.setText(musicSrv.getName());
    if(playbackPaused){
        try{
        setController();
        }catch (Exception e){

        }
        playbackPaused=false;
    }

try{
            controller.show(0);
        }catch (Exception e){}

    }


    public boolean onCreateOptionsMenu(Menu menu) {


        getMenuInflater().inflate(R.menu.menu_player_main, menu);
        return true;
    }


    public boolean onOptionsItemSelected(MenuItem item) {




        switch(item.getItemId())
        {
            case R.id.action_end:
                        stopService(playIntent);
                        musicSrv=null;
                        System.exit(0);
                break;
            case R.id.action_shuffle:

            case R.id.action_refresh:
                Toast.makeText(PlayerMain.this,"Updating
Song",Toast.LENGTH_SHORT).show();
                Toast.makeText(PlayerMain.this,"Songs
Refreshed",Toast.LENGTH_SHORT).show();
                onCreate(Bundle.EMPTY);
                break;
            case R.id.sensor_switch:
                if(item.isChecked()){
                    item.setChecked(false);
                    isChecked=false;
                    Toast.makeText(PlayerMain.this, "SENSOR CONTROL
OFF", Toast.LENGTH_LONG).show();
                    sm.unregisterListener(this,sensor);
                }
                else {
                    item.setChecked(true);
                    isChecked=true;
                    Toast.makeText(PlayerMain.this, "SENSOR CONTROL
ON", Toast.LENGTH_LONG).show();

sm=(SensorManager)getSystemService(SENSOR_SERVICE);

sensor=sm.getDefaultSensor(Sensor.TYPE_PROXIMITY);
                    sm.registerListener(this, sensor,
SensorManager.SENSOR_DELAY_NORMAL);
                }

                return true;
        }




        return super.onOptionsItemSelected(item);
    }
    public void getSongList(){

        ContentResolver musicResolver=getContentResolver();

        Uri musicUri=
android.provider.MediaStore.Audio.Media.EXTERNAL_CONTENT_URI;
        Cursor musicCursor=musicResolver.query(musicUri, null, null,
null, null);

        if(musicCursor!=null && musicCursor.moveToFirst()){
            int
titleColumn=musicCursor.getColumnIndex(android.provider.MediaStore.A
udio.Media.TITLE);
            int
idColumn=musicCursor.getColumnIndex(android.provider.MediaStore.Audi
o.Media._ID);
            int
artistColumn=musicCursor.getColumnIndex(MediaStore.Audio.Media.ARTIS
T);
            do {
                long thisId=musicCursor.getLong(idColumn);
                String thisTitle=musicCursor.getString(titleColumn);
                String
thisArtist=musicCursor.getString(artistColumn);
                songList.add(new Song(thisId,thisTitle,thisArtist));

            }
            while(musicCursor.moveToNext());
        }
    }
    protected void onStart(){
        super.onStart();
        if(playIntent==null)
        {
            playIntent=new Intent(this,MusicService.class);
            bindService(playIntent,musicConnection,
Context.BIND_AUTO_CREATE);
            startService(playIntent);
        }
    }
    public void songPicked(View view){

musicSrv.setSong(Integer.parseInt(view.getTag().toString()));


        musicSrv.playSong();

       textView.setText(musicSrv.getName());
        if(playbackPaused){
            setController();
            playbackPaused=false;
        }
        controller.show(0);
    }


    public void start()  {
        musicSrv.go();
    }


    public void pause()  {

        playbackPaused=true;
        musicSrv.pausePlayer();

    }


    public int getDuration()  {
        if(musicSrv!=null && musicBound && musicSrv.isPlay()){
            return musicSrv.getDur();
        }
        else
            return 0;
    }


    public int getCurrentPosition()  {
        if(musicSrv!=null && musicBound && musicSrv.isPlay()){
            return musicSrv.getPos();
        }
        return 0;
    }


    public void seekTo(int pos)  {
        musicSrv.seek(pos);
    }


    public boolean isPlaying()  {
        if(musicSrv!=null && musicBound){
            return musicSrv.isPlay();
        }
        return false;
    }
```

```
public int getBufferPercentage() {
    return 0;
}


public boolean canPause() {
    return true;
}


public boolean canSeekBackward() {
    return true;
}


public boolean canSeekForward() {
    return true;
}


public int getAudioSessionId() {
    return 0;
}
protected void onPause(){
    super.onPause();
    paused=true;
}
protected void onResume(){
    super.onResume();
    if(paused){
        setController();
        paused=false;
    }
}
protected void onStop(){
    controller.hide();
    super.onStop();
}
protected String songName(){
  return  "";
}


public void onSensorChanged(SensorEvent event) {
    timer=SystemClock.elapsedRealtime();






        if (event.values[0] == 0) {
                i += 1;
                t.cancel();
                t = new Timer();

                t.schedule(new TimerTask() {

                    public void run() {
                        if (i == 1) {

checkTimer=SystemClock.elapsedRealtime();
                            if(checkTimer-timer>1000  &&
isPlaying())
                            {

                                musicSrv.pausePlayer();
                                i=0;
                                PlayerMain.this.runOnUiThread(new
Runnable() {

                                    public void run() {

textView.setText(musicSrv.getName());
                                    }
                                });

                            }
                            else if(checkTimer-timer>1000 &&
!isPlaying()){

                                i=0;
                                start();


                            }
                            else {

                                musicSrv.playNext();
                                i = 0;
                                PlayerMain.this.runOnUiThread(new
Runnable() {

                                    public void run() {

textView.setText(musicSrv.getName());
                                    }
                                });
                            }

                        }
                        if (i == 2) {

                            musicSrv.playPrev();
                            i = 0;
                            PlayerMain.this.runOnUiThread(new
Runnable() {

                                public void run() {

textView.setText(musicSrv.getName());
                                }
                                });

                            } else

                                i = 0;

                        }
                    }, 1000);
                }



        }

    public void onAccuracyChanged(Sensor sensor, int accuracy) {

    }
    public boolean onPrepareOptionsMenu(Menu menu){
        MenuItem checkable=menu.findItem(R.id.sensor_switch);
        checkable.setChecked(isChecked);
        return true;
    }
```

MUSICSERVICE.JAVA

```
package com.example.hp.player;

import android.animation.AnimatorSet;
import android.app.Service;
import android.content.ContentUris;
import android.content.Intent;
import android.media.AudioManager;
import android.media.MediaPlayer;
import android.net.Uri;
import android.os.Binder;
import android.os.IBinder;
import android.os.PowerManager;
import android.provider.MediaStore;
import android.support.v4.media.MediaDescriptionCompatApi21;

import android.util.Log;

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

import java.util.Random;
import android.app.Notification;
import android.app.PendingIntent;
import android.widget.TextView;



public class MusicService extends Service implements
MediaPlayer.OnPreparedListener,MediaPlayer.OnErrorListener,MediaPlay
```

```
er.OnCompletionListener {
    private MediaPlayer player;
    private ArrayList<Song> songs;
    private int songPos;
    TextView currSongView;

    private final IBinder musicBind=new MusicBinder();
    private String songTitle="";
    private static final int NOTIFY_ID=1;

    public void onCreate(){
        super.onCreate();
        songPos=0;
        player=new MediaPlayer();
        initMusicPlayer();

    }
    public void initMusicPlayer(){
        player.setWakeMode(getApplicationContext(),
PowerManager.PARTIAL_WAKE_LOCK);
        player.setAudioStreamType(AudioManager.STREAM_MUSIC);
        player.setOnPreparedListener(this);
        player.setOnCompletionListener(this);
        player.setOnErrorListener(this);
    }
    public void setList(ArrayList<Song> theSongs){
        songs=theSongs;
    }
    public class MusicBinder extends Binder{
        MusicService getService(){
            return MusicService.this;
        }
    }
    public void playSong(){

        player.reset();

        Song playSong=songs.get(songPos);

        songTitle =playSong.getTitle();

        long currSong=playSong.getId();
        Uri trackUri=
ContentUris.withAppendedId(MediaStore.Audio.Media.EXTERNAL_CONTENT_U
RI,currSong);
        try{
            player.setDataSource(getApplicationContext(),trackUri);
        } catch (Exception e) {
            Log.e("Music Service","Error setting data source",e);
        }
        player.prepareAsync();
    }
    String getName(){

            Song song = songs.get(songPos);
            return song.getTitle();
```

MUSICCONTROLLER.JAVA

```
package com.example.hp.player;

import android.content.Context;
import android.util.AttributeSet;
import android.widget.MediaController;



public class MusicController extends MediaController {
    public MusicController(Context c)  {
        super(c);
    }
    public void hide(){

    }
}
```

SONG.JAVA

```
package com.example.hp.player;



public class Song {
    private long id;
    private String title;
    private String artist;
    public Song(long song_Id,String song_Title,String song_Artist){
        id=song_Id;
        title=song_Title;
        artist=song_Artist;
    }
    public long getId(){return id;
    }
    public String getTitle(){
        return title;
    }
    public String getArtist(){
        return artist;
    }
}
```

SONGADAPTER.JAVA

```
package com.example.hp.player;

import android.app.Activity;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import java.util.ArrayList;
import android.content.Context;
import android.view.LayoutInflater;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;



public class SongAdapter extends BaseAdapter {
    private ArrayList<Song> songs;
    public ImageView pos;
    private LayoutInflater songInf;
    public SongAdapter(Context c,ArrayList<Song> theSongs){
        songs=theSongs;
        songInf=LayoutInflater.from(c);
    }

    public int getCount() {
        return songs.size();
    }


    public Object getItem(int position) {
        return null;
    }


    public long getItemId(int position) {
        return 0;
    }


    public View getView(int position, View convertView, ViewGroup
parent) {
        LinearLayout
songLay=(LinearLayout)songInf.inflate(R.layout.song, parent, false);
        TextView
songView=(TextView)songLay.findViewById(R.id.song_title);
        TextView
artistView=(TextView)songLay.findViewById(R.id.song_artist);


        Song currSong=songs.get(position);
        int a=(int)currSong.getId();




        songView.setText(currSong.getTitle());
        artistView.setText(currSong.getArtist());
        songLay.setTag(position);
        return songLay;
    }
}
```

SONG.XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
android:layout_height="wrap_content"
    android:onClick="songPicked"
    android:orientation="vertical"
    android:padding="5dp">
<TextView
    android:id="@+id/song_title"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:textColor="#000000"
    android:textSize="18sp"
    />
    <TextView
        android:id="@+id/song_artist"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textColor="#000000"
        android:textSize="16sp"/>

</LinearLayout>
```

STRING.XML

```
<resources>
    <string name="app_name">Sensor Play</string>

    <string name="hello_world">Hello world!</string>
    <string name="action_settings">Settings</string>
</resources>
```

STYLES.XML

```
<resources>


    <style name="AppTheme"
parent="Theme.AppCompat.Light.DarkActionBar">

    </style>

</resources>
```

CONCLUSION

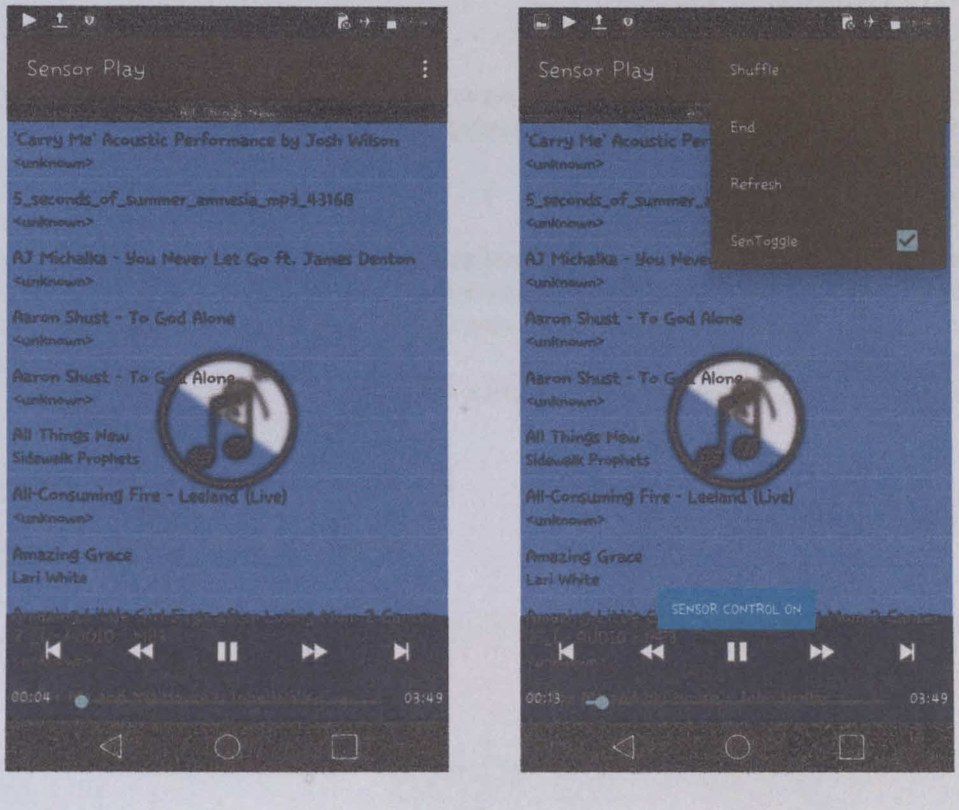
Finally,I have reached the concluding part of my project work and it has been a healthy and fruitful experiences for me.This project may be the stepping stone for me as a student to develop system apps and to keep on continuing and improve my skills in application development ,be it for any environment to run at.Before I started this project,most of the concepts were not clear to me and they are just like looking at alphabets by an illiterate person.But,with the completion of this project,my concepts have improved a lot.My confidence for building applications have been upped and will try my level best to build more and more projects like these in the near future.Hope this project will be a successful as for the part of satisfying user needs and there positive response will even boost up my confidence more.

BIBLOGRAPHY

Internet Sources:

- wikipedia
- androiddevelopers.com
- stackoverflow.com
- androidforums.com

  and some more websites.



'Carry Me' Acoustic Performance by Josh Wilson
<unknown>

5_seconds_of_summer_amnesia_mp3_43168
<unknown>

AJ Michalka - You Never Let Go ft. James Denton
<unknown>

Aaron Shust - To God Alone
<unknown>

Aaron Shust - To God Alone
<unknown>

All Things New
Sidewalk Prophets

All-Consuming Fire - Leeland (Live)
<unknown>

Amazing Grace
Lari White

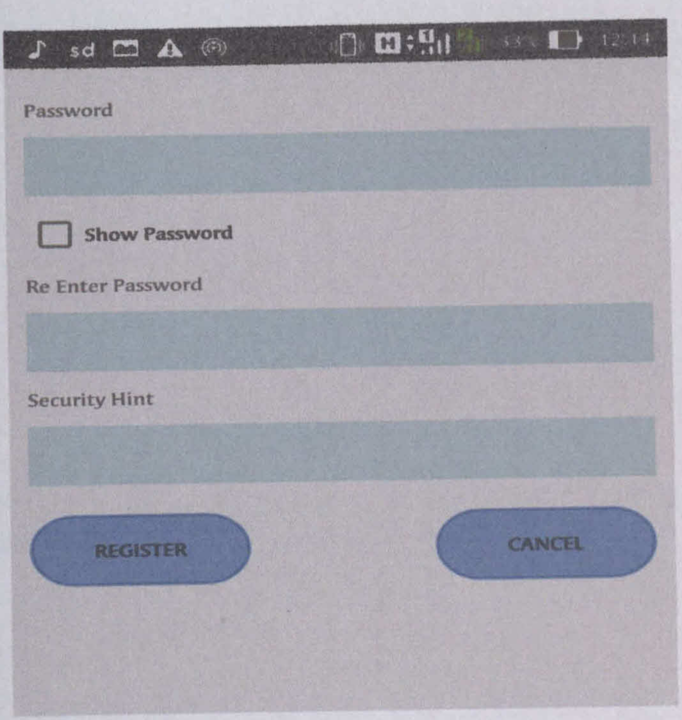
Output

```
                        startActivity(ii);
                    }
                });

            }

        }
```

output:



forget_search.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:background="#D8D8D8"
android:orientation="vertical" >
<RelativeLayout
android:layout_width="wrap_content"
android:layout_height="wrap_content"
>

<RelativeLayout
android:id="@+id/relativeLayout1"
android:layout_width="match_parent"
android:layout_height="wrap_content"
```