```
package com.internshala.echo.fragments
import android.app.Activity
import android.content.Context
import android.media.AudioManager
import android.media.MediaPlayer
import android.net.Uri
import android.os.Bundle
import android.support.v4.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.ImageButton
import android.widget.SeekBar
import android.widget.TextView
import com.internshala.echo.CurrentSongHelper
import com.internshala.echo.R
import com.internshala.echo.Songs
import java.util.*
/**
 * A simple [Fragment] subclass.
 */

class SongPlayingFragment : Fragment() {

    var myActivity: Activity? = null

    var mediaPlayer: MediaPlayer? = null

    /*The different variables defined will be used for their respective purposes*/
    /*Depending on the task they do we name the variables as such so that it gets easier to
identify the task they perform*/
    var startTimeText: TextView? = null
    var endTimeText: TextView? = null
    var playPauseImageButton: ImageButton? = null
    var previousImageButton: ImageButton? = null
    var nextImageButton: ImageButton? = null
    var loopImageButton: ImageButton? = null
    var shuffleImageButton: ImageButton? = null
    var seekBar: SeekBar? = null
    var songArtistView: TextView? = null
    var songTitleView: TextView? = null
    var currentPosition: Int = 0
    var fetchSongs: ArrayList<Songs>? = null

    /*The current song helper is used to store the details of the current song being
played*/
    var currentSongHelper: CurrentSongHelper? = null

    override fun onCreateView(inflater: LayoutInflater?, container: ViewGroup?,
                              savedInstanceState: Bundle?): View? {
        val view = inflater!!.inflate(R.layout.fragment_song_playing, container, false)

        /*Linking views with their ids*/
        seekBar = view?.findViewById(R.id.seekBar)
        startTimeText = view?.findViewById(R.id.startTime)
        endTimeText = view?.findViewById(R.id.endTime)
        playPauseImageButton = view?.findViewById(R.id.playPauseButton)
        nextImageButton = view?.findViewById(R.id.nextButton)
        previousImageButton = view?.findViewById(R.id.previousButton)
        loopImageButton = view?.findViewById(R.id.loopButton)
        shuffleImageButton = view?.findViewById(R.id.shuffleButton)
        songArtistView = view?.findViewById(R.id.songArtist)
        songTitleView = view?.findViewById(R.id.songTitle)
        return view
    }
    override fun onAttach(context: Context?) {
        super.onAttach(context)
        myActivity = context as Activity
    }

    override fun onAttach(activity: Activity?) {
        super.onAttach(activity)
        myActivity = activity
    }

    override fun onActivityCreated(savedInstanceState: Bundle?) {
        super.onActivityCreated(savedInstanceState)

        /*Initialising the params of the current song helper object*/
        currentSongHelper = CurrentSongHelper()
        currentSongHelper?.isPlaying = true
        currentSongHelper?.isLoop = false
        currentSongHelper?.isShuffle = false

        var path: String? = null
        var _songTitle: String? = null
        var _songArtist: String? = null
        var songId: Long = 0
        try {

            path = arguments.getString("path")
            _songTitle = arguments.getString("songTitle")
            _songArtist = arguments.getString("songArtist")
            songId = arguments.getInt("songId").toLong()

            /*Here we fetch the received bundle data for current position and the list of
all songs*/
            currentPosition = arguments.getInt("position")
            fetchSongs = arguments.getParcelableArrayList("songData")

            /*Now store the song details to the current song helper object so that they
can be used later*/
            currentSongHelper?.songPath = path
            currentSongHelper?.songTitle = _songTitle
            currentSongHelper?.songArtist = _songArtist
            currentSongHelper?.songId = songId
            currentSongHelper?.currentPosition = currentPosition
        } catch (e: Exception) {
            e.printStackTrace()
        }

        mediaPlayer = MediaPlayer()
        mediaPlayer?.setAudioStreamType(AudioManager.STREAM_MUSIC)

        try {
            mediaPlayer?.setDataSource(myActivity, Uri.parse(path))
            mediaPlayer?.prepare()
        } catch (e: Exception) {
            e.printStackTrace()
        }
        mediaPlayer?.start()
    }

    /*A new click handler function is created to handle all the click functions in the song
playing fragment*/
    fun clickHandler() {

        /*The implementation will be taught in the coming topics*/
        shuffleImageButton?.setOnClickListener({
        })
        nextImageButton?.setOnClickListener({
        })
        previousImageButton?.setOnClickListener({
        })
        loopImageButton?.setOnClickListener({
        })

        /*Here we handle the click event on the play/pause button*/
        playPauseImageButton?.setOnClickListener({

            /*if the song is already playing and then play/pause button is tapped
            * then we pause the media player and also change the button to play button*/
            if (mediaPlayer?.isPlaying as Boolean) {
                mediaPlayer?.pause()
                currentSongHelper?.isPlaying = false
                playPauseImageButton?.setBackgroundResource(R.drawable.play_icon)

                /*If the song was not playing the, we start the music player and
                * change the image to pause icon*/
            } else {
                mediaPlayer?.start()
                currentSongHelper?.isPlaying = true
                playPauseImageButton?.setBackgroundResource(R.drawable.pause_icon)
            }
        })
    }

    /*The playNext() function is used to play the next song*/
    fun playNext(check: String) {

        /*Let this one sit for a while, We'll explain this after the next section where we
will be teaching to add the next and previous functionality*/
        if (check.equals("PlayNextNormal", true)) {
            currentPosition = currentPosition + 1
        } else if (check.equals("PlayNextLikeNormalShuffle", true)) {
            var randomObject = Random()
            var randomPosition = randomObject.nextInt(fetchSongs?.size?.plus(1) as Int)
            currentPosition = randomPosition
        }
        if (currentPosition == fetchSongs?.size) {
            currentPosition = 0
        }
        var nextSong = fetchSongs?.get(currentPosition)
        currentSongHelper?.songPath = nextSong?.songData
        currentSongHelper?.songTitle = nextSong?.songTitle
        currentSongHelper?.songArtist = nextSong?.artist
        currentSongHelper?.songId = nextSong?.songID as Long
        mediaPlayer?.reset()
        try {
            mediaPlayer?.setDataSource(myActivity, Uri.parse(currentSongHelper?.songPath))
            mediaPlayer?.prepare()
            mediaPlayer?.start()
        } catch (e: Exception) {
            e.printStackTrace()
        }
    }
}
```