SoundBoardItem.java

```
package com.capstone.ocelot.SoundBoardActivities;

import java.io.Serializable;

import android.content.Context;
import android.media.MediaPlayer;
import android.net.Uri;
import android.speech.tts.TextToSpeech;

public class SoundBoardItem implements Comparable<SoundBoardItem>, Serializable{

    private static final long serialVersionUID = -4944390099592827522L;
    private long mItemId = 0;
    private String mDescription;
    private int mSoundResourceId = -1;
    private String mSoundResourceLocation;
    private int mIconResourceId = -1;
    private Uri mImageResourceLocation;
    private boolean mHasImage = false;
    private boolean mHasSound = false;
    private boolean isExternalAudio = false;
    private boolean isExternalImage = false;
    private long numPlays = 0;

    transient protected Context mContext;
    transient protected TextToSpeech ttsPlayer;

    public SoundBoardItem(Context context, String description) { mDescription = description;
mContext = context; } //You must have a description to create a SoundBoardItem
    public void setItemId(long itemId) { mItemId = itemId; }
    public boolean hasImage() { return mHasImage; }
    public boolean hasSound() { return mHasSound; }
    public boolean isExternalSound() { return isExternalAudio; }
    public boolean isExternalImage() { return isExternalImage; }

    public void setExternalSound(boolean isExternal) { isExternalAudio = isExternal; }
    public long getItemId() { return mItemId; }
    public void setDescription(String description) { mDescription = description;}
    public String getDescription() { return mDescription; }
    public void setIconResourceId(int id) { mIconResourceId = id; mHasImage = true; }
    public void setIconResourceId(Uri uri) {
        isExternalImage = true;
        mHasImage = true;
        mImageResourceLocation = uri;
    }
    public int getIconResourceId() { return mIconResourceId; }
    public Uri getIconResourceUri() { return mImageResourceLocation; }
    public void setSoundResourceId(int id) { mSoundResourceId = id; isExternalAudio = false;
mHasSound = true;}
    public void setSoundResourceId(String location) { mSoundResourceLocation = location;
isExternalAudio = true; mHasSound = true; }
    public int getSoundResourceId() { return mSoundResourceId; }
    public String getSoundResourceLocation() { return mSoundResourceLocation; }
    public Long getNumPlays() { return numPlays; }
    public MediaPlayer getMediaPlayer(Context context){
        numPlays++;
```

```java
         //Return the MediaPlayer
        if (isExternalSound())
            return MediaPlayer.create(context, Uri.parse(getSoundResourceLocation()));
        else
            return MediaPlayer.create(context, getSoundResourceId());
    }
    @Override
    public int compareTo(SoundBoardItem another) {
        return another.getNumPlays().compareTo(this.getNumPlays());
    }
}
```