

迭代评估报告

评估日期：2021.5.4

组号	14	项目名称	基于 Android Studio 的手机 Karaoke 软件开发
迭代名称	技术原型迭代	实际起止日期	2021 年 3 月 29 日 - 2021 年 5 月 4 日
<p>任务达成情况：</p> <p>1. 已经完成的任务</p> <p>1) 管理员系统：管理员能够登陆管理员系统，进行上传歌曲信息和相关文件、浏览曲库、编辑或删除歌曲信息等操作；管理员上传文件后，后端将运行人声伴奏分离算法与基频分析算法，产生歌唱打分所需文件，并将其一起保存至服务器。</p> <p>2) 手机APP与后端的通信：在APP读取歌曲信息时，数据从服务器获取，文件也从服务器下载。下载的歌曲文件会被保存在本地，无需每次都反复下载相同的文件。</p> <p>3) 伴奏演唱模式：在选择一首歌曲后，用户可以选择伴奏演唱模式，跟随伴奏开始K歌。</p> <p>a. 音频录制：用户可以佩戴耳机进行录音，录音采样率为44.1KHz，单声道，比特率为每次采样16bit。如果用户不佩戴耳机，则手机播放的声音也会被录制。</p> <p>b. 暂停/继续演唱：用户在演唱的过程中可以自由地暂停/继续录音。</p> <p>c. MV播放：在演唱时，屏幕上方会播放歌曲的MV。</p> <p>d. 歌词高亮显示：在MV下方逐句显示高亮的歌词，高亮方式目前为每句内匀速高亮。</p> <p>e. 歌唱打分：用户每演唱完一句话，系统就会对用户的演唱进行实时打分，并显示出打分结果</p> <p>f. 开启/关闭原唱：用户可以随时开启/关闭原唱。</p> <p>g. 重唱：用户可以随时重新开始演唱。</p> <p>h. 段落演唱：用户可以随时结束演唱，不需要等待整首歌曲结束。</p> <p>4) 作品生成：在伴奏演唱模式结束后（歌曲结束或用户选择结束）用户可以对录音音频进行调整。</p> <p>a. 音量调节：用户可以调节伴奏和人声的音量。但是只能减小音量，不能在原来的基础上增加音量。</p> <p>b. 人声对齐：在录音的过程中会不可避免地产生1秒以内的延迟，用户可以通过拖动“人声对齐”进度条手动将录音与伴奏对齐</p> <p>c. 作品保存：用户可以将录音与伴奏合成并保存至本地。同一个作品不会被反复保存。</p> <p>d. 重唱：若用户对演唱不满意，可以返回伴奏演唱界面进行重唱。</p> <p>5) 本地录音管理：用户可以在本地录音管理界面浏览、播放、删除本地录音。</p> <p>6) 稳定性提升：应用对各种错误都有相应的提示信息及处理方法，除非管理员上传文件出错或文件大小过大基本不会出现卡死、闪退的问题。</p> <p>7) 歌唱打分：实现了人声伴奏分离，旋律提取和演唱准确性评分功能，打分对专业歌手，普通歌手，跑调的歌手均区分良好。</p> <p>2. 未完成的任务</p> <p>1) 自弹自唱算法设计：自弹自唱部分的开发尚未开始</p> <p>3. 任务完成情况评估结果</p> <p>在进度方面，本项目完成了迭代计划中优先级最高的基本功能，使软件已经具备可用性；但优先级较低的拓展功能尚未开发。</p> <p>在质量方面，已经完成的功能具有较高的稳定性和易用性，不会造成系统严重卡顿，界面简洁。</p>			

评审/测试的结果:

1. 稳定性测试

1) 伴奏演唱

- a. 在录音时使用暂停/继续功能不会出现录音断裂, 或暂停时仍在打分。
- b. 在录音时退出软件, 录音和伴奏的播放会暂停, 不会出现录音断裂、录音或打分仍在继续。
- c. 在任何时候选择重唱完成均不会使软件闪退或卡死

2) 文件下载

- a. 在文件大小过大或网速过慢时, 文件下载会因超时而中断, 显示错误信息并删除不完整的文件, 不会使软件无响应或出错。

2. 延迟测试:

- 1) 同时播放多段音频: 在同时播放原唱/伴奏, 录音/伴奏时, 延迟低于80毫秒

- 2) 录音: 用户录音相较于伴奏有0.2到1秒的延迟

- 3) 歌唱打分: 在一句话演唱完毕后, 对该句的打分需要等待1至2秒才会显示

3. 打分准确度测试

在使用艺术唱腔演唱时, 打分一般在90分以上; 演唱旋律准确但音色普通时, 打分一般在80到90; 严重跑调时打分一般低于70。作弊或者环境吵闹会让打分的平均分会降低10分左右。

问题、变更和返工:

1. 问题

- 1) 由于歌唱打分功能属于计算密集型任务, 对每句话打分有0.5至2秒的延迟。
- 2) 由于歌唱打分需要将音频切分成多段, 最后将所有录音合并为一个文件时可能导致人耳几乎无法分辨的音频损失。

2. 变更和反攻

在本迭代中项目的开发没有需要变更或返工的部分。

经验和教训:

1. 在多人合作开发时, 要先明确各自提供的接口以及负责部分的预期效果, 才能避免合并时的大量重写工作。
2. 在测试软件时, 要考虑到各种使用情况及边界条件; 除了测试功能性需求, 还要测试非功能性需求, 例如是否有未释放的资源、无法结束的线程等。
3. 开发过程中要严格按照开发进度进行开发, 不能因为自己的任务可以在短时间内完成就推迟开发, 影响团队的开发进度。