

迭代评估报告

评估日期：2021.3.22

组号	14	项目名称	基于 Android Studio 的手机 Karaoke 软件开发
迭代名称	界面原型迭代	实际起止日期	2021 年 3 月 5 日 - 2021 年 3 月 22 日
<p>任务达成情况：</p> <p>① 已经完成的任務包括：</p> <p>1. 明确用户需求，确定项目功能</p> <p>(1) 用户需求：使用软件进行手机 K 歌，增加手机 K 歌软件的娱乐性，削减手机 K 歌软件中的冗余功能，还给用户一个轻量、纯粹的 K 歌体验。</p> <p>(2) 项目功能：歌曲在线搜索与下载，MV 画面伴奏演唱，自弹自唱，歌唱打分，本地录音播放与分享。</p> <p>2. 选择语言、工具及框架</p> <p>本项目选择使用 Android Studio 作为开发平台，使用 Java 与 Python 作为主要开发语言，使用 Vue 作为 Web 前端框架，使用 Python 的 Flask 框架与 MongoDB 数据库作为后端进行开发。</p> <p>3. 可行性分析</p> <p>(1) 歌唱打分：采用主流的有参考打分算法，即提取用户歌唱的旋律线，与原唱的旋律线进行比对。pyin 及其后续项目是常用的旋律提取算法，可实现音频到 midi 的转换。midi 文件中存储着每个音符的音高和时值，可设计算法比较用户与原唱的 midi 文件并根据差异大小进行打分。</p> <p>(2) 歌曲播放及录音：Android Studio 开发平台有对应的库组件能够实现音频的录制与播放。</p> <p>(3) 文件储存：MongoDB 数据库可以储存较大的文件（图片、音频、视频等），适合作为本项目的后端数据库。</p> <p>(4) 交互：MongoDB 数据库与 Android Studio 以及 Flask 框架都具有良好的交互性。</p> <p>4. 设计界面并实现界面原型</p> <p>本项目共包括两部分：用户手机端 app 部分与管理员 Web 端歌曲上传系统部分。目前为止已经实现的界面原型包括：</p> <p>(1) 用户手机端 app 部分：app 首页，伴奏演唱页面，自弹自唱页面，k 歌结果页面，歌曲搜索界面，本地录音浏览界面，用户个人界面，设置界面。</p> <p>(2) 管理员歌曲上传系统部分：登录页面，歌曲详细信息上传页面。</p> <p>5. 编写迭代评估报告以及 vision 文档，完成用例建模</p> <p>② 已经实现的功能包括：</p> <p>1. 手机端与 Web 端各自页面之间的跳转。</p> <p>2. 歌曲搜索。</p> <p>3. app 内歌曲播放、暂停、重放，歌曲音量调节，进度条拖拽。</p> <p>4. app 内 mv 与歌曲同步播放。</p> <p>5. k 歌界面歌词逐字高亮显示。</p> <p>6. 管理员歌曲上传系统登录验证。</p>			

③ 在进度方面，本项目目前进度达到预期。

④ 在质量方面，本项目界面美观，各元素布局合理，但在程序的鲁棒性上还有待提高，具体表现为有时会出现页面闪退。

评审/测试的结果：

经过初步的人工测试，本项目虽然实现了上述功能，但仍然存在下述问题。除此之外，本次迭代完成了迭代计划中安排的任务。

问题、变更和返工：

① 问题：

1. 目前歌词高亮功能为匀速高亮，而实际上一首歌曲每句歌词播放的速度是不定的，因而并未实现逐字同步。
2. 歌曲播放时，进度条有时不能正确地显示播放时间。
3. 当歌曲文件大小过大时，可能无法正常同时播放多个音频文件。
4. 在退出并重新打开软件时，会发生闪退。

② 变更：本项目目前暂时无变更。

③ 返工：目前 app 播放的歌曲皆为本地音频，在之后的迭代中需要从服务器上获取歌曲播放资源进行播放。

经验和教训：

- ① 善于搜索和使用已有的库组件来实现对应功能，以提高开发效率。
- ② 实现每一个功能之后立即进行单元测试，及早发现问题所在并进行修正，避免其影响到之后的开发。
- ③ 在进行开发之前先进行项目整体架构的规划，再由整体到局部，进行细节的规划。