**迭代评估报告**

　　　　　　　　　　　　　　　　　　评估日期：2021-3-22

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 第十五组 | 项目名称 | 交我唱APP |
| 迭代名称 | 界面原型迭代 | 实际起止日期 | 2021-2-26~2021-3-23 |
| 任务达成情况： 1静态前端代码1.1完成的任务简述  |  | | --- | | 1.手机注册界面。 | | 2.动态页面。 | | 3.发现页面。 | | 4.K歌页面。 | | 5.K歌结算界面。 | | 6.我的界面 | | 7.我的歌曲界面。 | | 8.我的信息界面。 |   222  图 1 交我唱APP静态前端页面 1.2实现的功能  |  |  |  | | --- | --- | --- | | 功能 | 进度 | 结果质量 | | 手机号输入以及短信验证码校验 | 手机号可以正常输入；发送验证码需要接入付费短信平台，原理可行，暂时未实现；输入验证码后可以成功跳转。 | 较好。 | | 音乐播放器 | 可以根据本地资源路径或URL正确播放资源；已实现继续/暂停按钮；已实现上一首/下一首按钮；歌词显示界面可以解析LRC歌词文件，并且在对应的时间高亮当前歌词；实现了可以拖动的进度条。 | 质量尚可，进度好于预期。 | | 视频播放器 | 可以根据本地资源路径或URL正确播放资源；已实现继续/暂停按钮。 | 较好。 | | 底部导航栏 | 五个主功能界面通过底部导航实现切换；在需要全屏以提高用户体验的界面则隐去，使用顶部的返回导航作为替代。 | 较好，可以流畅切换界面。 | | 动态 | 展示样例的作品；引入跳转动作，用户点击后可以转到播放页面播放作品；引入动画效果，推荐页面呈卡片式，用户可以拖动来进行切换。 | 较好。 | | 发现 | 提供搜索栏供用户查找歌曲，提供部分推荐曲目。 | 搜索栏静态实现，未导入数据。 | | K歌 | 引入音乐播放器，包括曲目封面、曲目信息、可控的进度条和控制组件、滚动播放歌词。  静态展示切换原唱、返听调音、重录功能。导向完成K歌的结算页面。 | 页面及跳转切换基本实现。 | | 结算 | 由K歌页面进入，包括后期音效处理（未实现），和生成作品按钮，导向我的页面。实际音乐处理相关内容未部署。 | 跳转逻辑基本实现。 | | 消息 | 搭建了消息页面的静态框架，包括评论和点赞、礼物、最新听众。 | 页面搭建基本实现。 | | 我的 | 显示用户账号个人信息，包括粉丝、关注、好友等社区关系。  包含“我的作品”按钮，导向作品页面。 | 页面显示完整，跳转基本实现。 | | 作品 | 显示高赞作品和所有作品列表，信息包含封面、名称、创作时间、点赞数。 | 较好，页面显示基本实现。 | | UI风格 | 使用底部导航栏进行规划分页，使用了flexbox进行整体布局。美术风格为使用蓝色系纯色作为主题的简洁风格，图标等则使用统一的彩色底白色团。 | 较好，整体页面简洁美观。 | | 代码风格 | 使用了组件式开发。 | 由于第一次多人合作编程，代码风格未完全统一，变量命名和文件命名没有按照规范，成果不佳。在第二轮技术原型迭代时会顺带解决。 |  2前端复现教程2.1完成任务简述  |  | | --- | | 1.完成了安卓真机前端过程笔记，其他组员可以比较便捷地使用安卓真机上手前端配置和开发。 | | 2.完成安卓模拟器前端过程笔记，其他组员可以比较便捷地使用安卓模拟器上手前端配置和开发。 |  3后端demo框架和前后端通信3.1完成任务简述  |  | | --- | | 1. 选购了适合本次项目使用的腾讯云服务器。 | | 2. 在云端搭建了Lamp环境，通过apache实现了其服务器功能，从外界能够访问。 | | 3. 学习了php语言，并用它进行了后端的搭建。 | | 4. 学习并应用了ajax的方法，进行了前后端通信。 | | 5. 创建了数据库，并成功连接。 | | 6. 对访问后端的信息进行日志记录。 | | 7. 对于一部分访问异常的情况，能够反馈错误信息给前端。 | | 8. 了解并成功连接到网易云API。 |  3.2实现的功能、进度和质量  |  |  |  | | --- | --- | --- | | 功能 | 进度 | 结果质量 | | 云服务器搭建 | 完成了所需开发环境的配置，实现了从外部的正常访问。 | 较好。 | | 前后端通信 | 通过ajax传输数据，双向的通信都能成功进行，能够传输一个Java的对象，并能成功读取使用。 | 较好。 | | 后端数据库存储 | 配置好了数据库权限，在云服务器上搭建好数据库并能成功连接上。 | 功能基本实现，但还需后续实际使用中，更多的读写数据库测试来验证。 | | 访问日志记录 | 通过脚本，实现了对接受到的访问请求的主机、时间、用户名等相关信息进行记录。 | 功能基本实现，但随着后续的完善，会有更多的日志需要记录。 | | 错误反馈 | 对于访问异常的情况，反馈给前端错误产生的原因。 | 有待完善，能反馈一部分的情况的错误，其他情况还有待后续完善。 | | 网易云API连接 | 能通过网易云的接口，获取需要的音视频信息。 | 功能基本实现，但还需要后续发送给前端，测试是否也能正常使用。 |  4后端复现教程4.1完成任务简述  |  | | --- | | 1.服务器环境搭建：给出了腾讯云官方的云服务器环境搭建教程。 | | 2.前后端通信：给出了前端封装好的js函数及其调用方法，后端的php代码示例。 | | 3.日志记录和错误反馈：给出了实现的php代码示例。 | | 4.数据库连接：给出了权限设置命令，设置后即可正常连接。 | | 5.网易云API调用：给出了实现的php代码。 |  5音频编码学习及其对应教程5.1完成的任务和进度  |  |  | | --- | --- | | 任务 | 进度 | | 了解基本概念，包括采样、量化、编码等。 | 已完成。 | | 了解音频编码主要格式，包括WAV，MP3，AAC和OGG，以及相关编码工具。 | 已完成。 | | 学习配置第三方库的使用环境，了解交叉编译的原理，学习通过编写makefile文件编译第三方库。 学习通过JNI调用原生方法。 | 已完成，项目中暂时用不上 | | 学习在react native中使用ffmpeg模块。 | 已完成环境搭建与demo测试，尚未结合实际需求深入研究。 | | 学习Android平台音频采集的相关知识。 | 完成理论学习，学习了他人的demo，尚未自己上手编写代码。 | | 学习混音效果器的原理与实现，以及其相关数学知识和乐理知识。 | 正在学习中。 | | 针对伴奏获取困难的问题，研究现有的人声消除算法。 | 尚未找到好的开源解决方案，有了解的朋友可以推荐一下。 | | 给出了用php调用网易云api的代码示例。 | 实验成功，已移交后台处理。 | | 给出了使用ndk交叉编译第三方库方法的参考资料。 | 在自己的环境中实验成功。 | | 给出react native 中使用ffmpeg模块方法的demo。 | 在自己的环境中实验成功，尚未与前端对接。 | | 给出了在安卓中调用麦克风录音方法的代码示例。 | 来自网络的教程，尚未进行实验。 |  5.2质量评价 学习工作稳步进行，现有的成果应该已经可以使小组成员快速上手音频开发。但是仍有部分技术难点尚未攻破，有待进一步学习。 6软件测试原理学习及其教程6.1完成的任务和进度  |  |  | | --- | --- | | 任务 | 进度 | | 了解基本概念，包括原理，测试标准，测试工具等。 | 完成。 | | 根据各项标准，选择不同的工具，包括自动化测试appium-selenium，稳定性测试monkey，CPU，内存等系统测试adb等。 | 完成。 | | 学习配置测试工具的环境，掌握与Android交互的基本原理，学会用python控制appium的代码。 | 完成。 | | 给出appium的配置方法，以及对安卓手机进行自动化测试的demo代码。 | 完成。 |  6.2质量评价 自动化部分基本完成，现有成果可以支持软件自动化测试。稳定性与系统性能测试等还有待学习，在之后的后端测试里会有体现。 | | | |
| 评审/测试的结果： 1评审1.1内部评审 在软件开发的过程中，我们持续在项目组内定期召开内部评审会，对前一段时期的软件本身及其衍生文档进行评审。 1.1.1技术方案评审 在每周的例行组会中，我们会召开技术方案评审。针对本迭代的最终目标和当前的问题，评审当前的方案。主要针对方案主要能够解决的问题、可落地性、实现代价、是否要引入新组件、扩展性等方面进行内部评审，最终对提议的方案给出一个反馈意见。 1.1.2需求评审 在软件前景文档和软件需求规约每一个版本完成后，本项目组在微信群中立即召开非正式需求评审，要求项目组每一个成员检查文档中的每一个需求，每一行文字，每一张图标，并且从正确性、清晰性、无二义性、一致性、必要性、完整性、可实现性、可验证性、可测性等指标形成反馈意见。 1.1.3代码走查 代码走查可以提高软件的质量，以及可维护性。这样就可以减少查找错误的时间，提高解决bug的效率，提高开发效率的同时降低后期的维护成本。  本APP在本轮迭代中分为前端组和后端组进行分别开发。组内成员共同进行一个或多个功能或模块的开发。在合作开发的过程中，每个项目组成员自动充当代码走查的评审员，项目组成员需要保证其他人写的代码是可读易懂的。如果出现了难以理解的混乱的代码，需要及时给出评审意见，将意见反馈给那位写出晦涩代码的成员。 1.2同行评审1.2.1正规检视 我们充分利空课后时间，邀请沈备军老师对我们交我唱APP的迭代计划、Vision文档、用例模型等进行同行评议，给予我们一些改进的建议，同样地，助教老师对我们每周提交的软件相关文档进行正规检视后，会对软件的文档给出一个同行评审建议。基于此，我们再召开内部小组会对反馈内容进行消化和吸收，应用到之后的工程中去。 2界面原型测试2.1测试目的 测试前端静态原型的布局和UI是否合理美观、测试界面的用户友好程度、测试已实现的前端交互产生的结果是否符合用户的心理期待等。 2.2测试结果 按钮点击手感：基本上所有按钮点击后都会有颜色深浅的渐变过程。从渐弱到渐强，增加了点击的手感。少数按钮还未添加此手感。例如主页面点击“完成“后左下角和右下角的按钮，点击后并无反应，让人无法区分是点到了没反应还是根本没点到，有一些不舒适。  侧滑卡片有限：动态里实现了向两边划便可切换下一首歌的功能，类似于网易云等音乐软件。这个侧滑的手感较好。按业务逻辑来说不管侧滑多少次，都应从数据库里随机到下一首。而现在由于与数据库还未实现通信，本地歌曲数量有限，因此侧滑7次后界面便变为  空白，无法再切换歌。这里有待改进。  动态里的四个按钮都还未实现：可以设计成类似web一样的平行窗口。  滑动不畅：录歌界面，在播放歌曲时，歌词无法上下滑动，会迅速被它拉回正在唱的歌词，极为不方便。应当设计为，几秒不滑动再拉回正在唱的歌词。  网络：联网时可以播放歌曲，不联网没法播放歌曲。  测试版demo问题：下方会有出现的warning，影响用户体验。  按键完整程度：在不同的机型下，会出现某些按钮不兼容的情况。例如在某台华为手机上，歌曲暂停键点了以后，歌曲被暂停。此时，播放键消失。它虽然依然在原先的位置，可以被点击，但我们却看不见，初步判断是个兼容性的bug。  图片加载：绝大多数图片可以加载，少部分图片无法加载。例如点击“我的作品”按钮后，作品左侧的封面无法被加载。  歌曲社区功能：  视频播放：动态页面的歌曲应当可以点进去播放视频。但没连接到电脑上时，视频无法正常播放。分析原因是视频还未作为资源插入程序，现在只是保存在电脑本地，因此只有连接电脑时可以访问。  界面重叠：动态页面下，上面四个“全部“”热门“”好友“”教唱“按钮被挡住，无法点击。点击时会自动判定为拖拽中间的歌曲。只有等到中间歌曲拖拽完变为空白时，才能点击上方按钮。而实际情况应当是中间会是歌曲流，永远不会停下，这会导致上方按钮永远无法点击。  测试共发现1个严重错误，3个一般错误，6个建议改进问题。 | | | |
| 问题、变更和返工： 1前端的问题、变更和返工 **问题1**：最初页面设计过于单薄、使用简单的按钮以及导航组件进行导航。页面整体显得杂乱且跳转切换的管理成本较高，需要改善。  **变更**：重新制定了页面逻辑和需求功能对应的页面呈现。改为使用统一的底部导航Tabbar进行导航，与原有的导航组件相组合跳转子页面。  **返工**：需要返工。重新制作了前期的部分页面，将新页面添加进逻辑。取消了原本用按钮管理全部导航跳转的方式并引入底部导航组件。  **问题2**：图片样式的引入和设定过于复杂、体积较大且难以达到本项目ui风格的要求。  **变更**：使用Svg的方式来代替图片插入作为绝大多数按钮类选项的解决方案。  **返工**：将原本使用的图片按钮替代为Svg和文本的组合。 2后端的问题、变更和返工 **问题1**：服务器搭建问题，由于云服务费用较贵，一开始尝试的是在本地用自己的电脑搭建服务器，结果只能局域网内连接服务器，学校也不允许进行公网ip的请求访问，不能达到项目需要的后端服务器需求。  **变更**：由本地电脑搭建服务器改为云服务器，找到了通过学生优惠租借服务器，费用在项目承受范围内。  **返工**：需要返工，重新搭建服务器环境。  **问题2**：一开始商议决定使用spring及其衍生的架构来进行后端环境的搭建，因为这是目前最主流的实现方案之一，功能较为全面系统，参考的资料也很多。但当我们开始学习相关内容时，发现spring架构内容十分多，架构体系非常庞大，功能多却也较为复杂，再加上我们都没有java语言的基础，还需要先补习java，学习成本较高，入门较为困难。  **变更**：就此进行了开会讨论，我们认为spring由于功能强大架构庞大，更加适合一些较为大的项目比如一些企业级项目，由于迭代计划要做出后端demo，目前看来学习成本有些超出预期。于是我们重新审视了一下本次项目对于后端的要求，需要实现的功能主要就分为两部分：记录用户的基本信息和调用网易云的api后传输数据给前端，并不需要实现很多复杂的功能，用spring有些大材小用了。最后我们决定改为使用php语言进行后端开发，它脚本语言的特性，能更加快速地实现一些需要的功能，更容易入门，用它能快速搭建出后端demo，更重要的是，它也完全能够胜任本项目的需求。  **返工**：还没有开始做就及时改变了计划，所以无需返工，但也花费了一些了解学习spring的时间，导致进度风险加大。  **问题3**：服务器端搭建好了数据库环境并能创建数据库，但是用php语言去调用数据库时，无法正常连接进行读写。  **变更**：由于一开始找不到问题原因，暂时用读写文本文件的方式来替代数据库；后来找到了出错原因，是数据库的访问权限没有正确设置，更改后问题解决。  **返工**：无需返工，仅用了一段时间的替代方案。 | | | |
| 经验和教训：   1. 规划页面的时候，不能走一步看一步，因先设计好整体的逻辑框架，调查可能排得上用处的组件作为备选项再开始动工，可以大大减少需要返工的情况。 2. 具体操作实践时需要引入已经实现的react-native相关组件，此时需要注意组件的依赖是否安装完整并充分阅读该组件的使用文档，以防加入代码后无法运行的情况。 3. 项目开发过程中需要及时备份，并准备多个版本，以防引入新的变动时整体崩溃无法回溯的情况。 4. 选取所用的技术栈时，不能盲目随大流，应当分析清楚自己的功能需求和进度安排，充分考虑功能上的完备性和进度上的可行性后，再根据不同技术方案的特性进行选择。 5. 当碰到一些一时无法解决的报错时，不要跟一个环节死磕，需要积极寻找替代方案，使得后续的任务能够进行下去，计划不被耽搁。 6. 当碰到问题时，应当及时进行组内沟通，如果是原计划出现了问题，就应当及时进行修正或更改，才能不耽误进度；否则若是害怕返工，一条路走到黑，会造成更大的损失。 | | | |