总产品实现方案

目录

1. 系统的主要功能

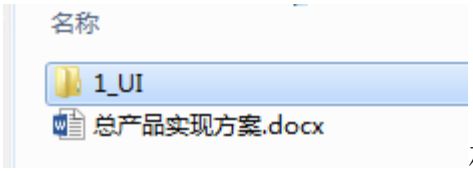
本平台为基于 AR 技术和网络爬虫的音乐聚合服务平台，下面简称“聚音乐平台”。顾名思义，我们平台提供聚合音乐信息,除此之外还提供相应娱乐功能。目前我们已经完成了最基本的搜索功能和娱乐功能，包括三端，后端的信息爬取，筛选，格式化；安卓端的信息搜索与展示，进一步还有相关的链接跳转功能；PC 端相关网页开发，包括相应的 UI 设计，显示的相关逻辑功能，搜索查询功能。

功能模块	子模块	内容说明	优先级	完成程度
个人中心	信息	提醒收到的私信、评论、点赞	中	个人中心算作一个单独的模块进行开发。目前进度 60%，仍是单机版，逻辑设计完成了，包括个人信息，资料、好友、关注等行为，现在只差与后台进行数据交互
	每日签到	每日签到赚取积分	低	
	积分管理	每日签到、搜索歌曲、分享歌曲、玩游戏等都能够积累积分，可以使用积分抽奖或升级	高	
	个人资料	更换用户头像、签名、昵称	高	
	我的好友	可查看粉丝和关注	中	
	听歌计时	用户想看看今天一共听了多久的歌，以及自己的听歌类型分布，并且可以将结果以图片形式分享	中	

设置	主题		中	由于优先级较个人中心低，仍未处理，难度较低。
	夜间模式		中	
搜索	按歌手搜索	显示出该歌手的所有歌曲，用户点击某一歌曲后再转到按歌名搜索界面	高	搜索功能，即按歌名搜索已经实现了相应的功能了，我们预计划从六个平台获取歌曲信息，并通过筛选提供，尽量提供高质量的搜索结果，尽量一步到位，比如点击直接播放。但仍然有些问题没有处理，涉及到算法问题，比如，我们什么时候更新后台的数据？定期去更新还是用户查询再去更新数据库的数据，这里设计到响应速度与用户体验的问题。进度 70%。
	按歌名搜索	将搜索结果按最适合用户的结果显示出来，无搜索结果则推荐榜单	高	
热搜榜	-	显示各大音乐平台的热搜榜情况	较高	优先度低于搜歌功能，预计是三天之内完成。
评论猜歌	-	根据各大音乐平台上听众对音乐的评论，猜一猜是哪一首歌曲，有积分奖励	高	安卓端、WEB 端已经完成了相应的 UI 设计，并展示了 DEMO，现在还在考虑评论信息的来源。我们要尊重每个平台的信息保护策略，尽量合法的获取。进度 40%
分享	分享到第三方平台	将歌曲以图片的形式分享	高	分享功能已经完全实现了，进度 100%。可以分享到微信朋友圈等第三发平台 。
	分享到动态	发布到本平台中，可选择权限	中	
附近的人	搜索列表	可以知道附近的人搜索列表	中	优先度较低，仍未进行。进度 0%
	地图分布	显示搜索某首歌的人在地图中的分布	中	
AR 游戏	放置	打开定位，将歌曲放置在当前位置上	高	由于时间不足与难较大，预计在寒假继续

	捕获	通过 AR 扫描功能扫出并捕获歌曲，可以将结果分享	高	完成相应功能开发。进度 0%

2. UI 界面设计



相应完整的安卓端和 Pc 端 UI 设计已经放在同名目录下来了。在这里简单展示相应的 PC 展示

3. 关键技术和技术难点

关键技术：

1. 安卓：java+eventBus
2. 服务器：SpringMVC4.0+Hibernate+MySQL+python
3. Web 端：Angular+BootStrap

安卓端：

EventBus 消息队列	由于 App 的需求较复杂，为了提高开发效率，团队使用了在 Github 上开源的 20 余个第三方库，利用它们在 App 中构建了大量的子模块。然而，这些模块间应该如何通信成为了一个问题。我们起初在后台 Service 中使用 BroadcastReceiver 来发送和接收消息，但由于 BroadcastReceiver 存在历史遗留和兼容性问题，在不同 Android 版本下表现行为不一致，且由于一旦注册长期存在于全局空间，导致系统运行效率降低，出现耗电和 App 被重复唤醒的问题，而且如果开发者忘记取消注册会导致内存泄漏。除此之外，使用 BroadcastReceiver 使得模块间的耦合度很高，维护和扩展困难，在 App 开发过程中易出错，是一个不良的架构。为了解决这一问题，我们使用了 EventBus，这一工具是建立在 Android 上的消息队列，类似于 RabbitMQ 等 Server 端的消息队列，这一工具可以很好地取代 BroadcastReceiver 的功能。在函数式编程、订阅者的设计模式和诸多优秀架构的帮助下，实现了消息发送者和接收者的完全解耦，极大地提高了项目的可维护性。
高难度界面的实现	**1 顶部导航栏转成底部导航栏** 在需求分析、软件 UI 界面设计阶段，我们在 ProcessOn

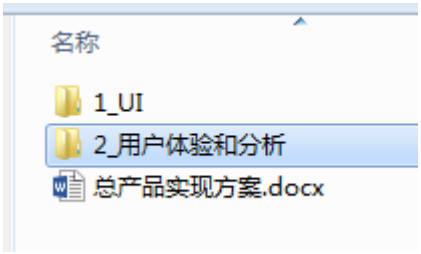
	上拟定了一套 App 的基础界面设计方案，主界面是以顶部导航栏为基础的，一开始我们用 TabLayout 嵌入顶部导航栏的方法实现了这个界面，也这基础上完成了很多设计。但后来我们发现这样设计不够符合 Material Design，经过百灵同学的再设计，我们打算把顶部导航栏换成底部导航，于是整个 App 的界面都要更改，我们在这里花了很多时间，App 的开发进度也因此拖延了。
	2 CardView 实现动态界面 动态界面需要使用 CardView 来实现，安卓 UI 界面的两位主设计者都是第一次接触 CardView，它是基于 RecyclerView 的基础上实现的，而在里面还要实现像点赞、转发、评论这些开发难度较高的功能，这期间出现了很多闪退困难 ，我们都一一解决了，目前还有不足的地方，日后会不断完善。
	3 SearchView 嵌入导航栏 SearchView 的使用也是一大难点，我们希望实现像“网易云音乐”搜歌界面那样的搜索栏，将搜索栏嵌入顶部导航栏。实现的过程和平时我们实现控件的过程有不一样的地方，不是简单地通过 findViewById 获取的，所幸通过查资料得到一个比较完整的教程，最终得以实现。

多个数据库的管理	由于 App 的难点较高，功能比较复杂，所以在服务器端要存储的信息就比价繁杂，除了用户的账号密码，还有用户头像、个性签名、关注我的人、我的关注、发表的动态、点赞数量等等，甚至还有我们仍未完成的等级、积分功能。由于数据量较大，所幸这些数据都可以用文本来表示除了头像，于是我们只给用户几种头像选择，不允许用户自定义头像，这样子的话头像信息也可以用文本来表示了。服务器端还需要存储的数据还有歌曲信息、歌曲评论、AR 地图信息（未完成）等等。在用户本地存储的数据有搜索的历史记录，账号信息。
服务器端与 Android 端的通信	**1 登录状态的维护** 在用户登录后，服务器端需要将登录凭证（SessionId）发送给 Android 端，Android 端的后续请求都需要附上这个 SessionId 以证明自己的身份。在此情况下，服务器需要维护一个登录状态表，以及一定的过期时间。部分接口在用户已登录和未登录状况下有着不同的行为，这些行为可以互相组合，相当于笛卡尔积，产生非常多的可能行为，极大地增加了程序的复杂度和代码量，是开发工作的挑战之一。
	2 高并发请求下的可用性保证 当在线用户数量较多时，服务器需要每秒处理数百次甚至数千次数据请求，此时服务器的压力很大，为了保证

	用户端的等待时延较低，需要对服务器进行动态扩容，和对爬虫的深度算法优化。当无法确保服务在一定时延内的可用性时，需要给用户良好的提示信息。除此之外，还需要防范针对服务器进行的 DDoS 攻击。
	3 传输安全 由于数据在传输过程中可能被黑客通过监听链路等方式盗取，导致用户信息被盗或隐私泄露，因此需要对数据进行加密或使用安全传输协议。为了解决这一问题，开发团队正在筹划部署 Let’s Encrypt HTTPS 安全证书

目前主要难点是法律问题，我们涉及到相关的音乐信息都具有版权纠纷，要做到尊重版权。

4.用户体验和分析



内容另存为文件夹 用户体验及分析

5.已完成的改进和存在的问题

已完成的改进	
1 搜索框的改进	经过对用户体验的分析，我们收集到很多非常

	有帮助的意见。对于搜索框，我们将它变成圆角型，使之不会显得这么突兀，与整个界面融洽。
2 动态界面的改进	之前我们的动态界面是基于 ListView 的，没有动态效果、点赞功能等，于是我们更改为 CardView 显示，增加转发、点赞功能并实现显示发布时间。
3EventBus 消息队列的使用	使用这个消息队列之前，我们实现登录功能是通过 BroadcastReceiver 来发送接收消息，后面出现了一些无法解决的 Bugs,比如登录成功后不会跳转到主界面。通过查阅资料了解到这种方法的种种弊端，并学到了新的管理消息队列方式，所以我们果断放弃它，使用 EventBus 消息队列。
4 搜歌的加载动画	增加搜索歌曲时的加载动画后，整个搜歌过程更加人性化，动画效果像是音符在跳动，正好符合了我们 App 的主题。
5 增添启动页	有用户反应我们的App要是增加启动界面会更漂亮，我们特地为此设计了一个精美的、符合主色调的启动界面，主界面上写着我们 App 名字。添加上去后App果然像是提升了档次一样，更像是市面上流通的 App，几行字在上面让用

	户能记住我们。
6 分享到其他平台	我们在听歌界面实现了向微信、微博等平台分享的功能，我们会随机生成一张精美的图片配上歌曲名字和作者，图片底部还有App的水印，精致漂亮。在以后希望能够实现在图片上配歌词。
7 互动界面的改进	App 初稿中动态界面是基于 ListView 的，界面单一，为了提高设计感，使用了更优的构图，界面由四个方块组成，增强了整体的美感与用户体验。

存在的问题	
1 歌曲分享到动态	不能将歌曲分享到动态，用户点击后可直接跳转听歌，这个功能虽然写在我们的需求里，但具体实现起来还比较困难。
2 评论功能	未实现评论动态的功能。如果真正要实现，应该是要在 CardView 上做一些改动.
3 头像的选择未实现	本来的计划是指定好 App 固有的十来种头像风格，用户只能从设定好的头像中选择，但由于时间有限，此功能没现。
4 设置里的功能待完善	在设置里有主题切换、夜间模式、清除缓存、意见反馈等等功能，但都因为时间不足且优先级较低而暂时放置一边了。
5 AR 功能未开发	AR 功能包括 AR 地图和扫一扫，都是我们 App 重要功能，但是 AR 功能实现难度真的很大，作为安卓开发的初学者，在完成了复杂界面设计和交互后，没有时间学习 AR 功能的开发，相信以后一定能够实现。