

**毕业设计（论文）**

**题目：基于Android的音乐App开发**

**学院：计算机科学学院**

**专业班级：软件工程1602**

**指导教师：王崑凌职称：讲师**

**学生姓名：李鹏飞**

**学号：41609050212**



**毕业设计（论文）**

**题目：基于Android的音乐App开发**

**学院：计算机科学学院**

**专业班级：软件工程1602**

**指导教师：王崑凌职称：讲师**

**学生姓名：李鹏飞**

**学号：41609050212**

# 摘要

在当今社会，互联网已经成为人们日常生活中密不可分的一部分，越来越多的信息通过互联网展示在我们眼前。人们越来越趋向于集成化的的信息获取方式，对于喜好音乐的听众和音乐人来说，则需要一个便捷的渠道来获取音乐信息。因此有必要开发一款符合当前用户需求的音乐类手机软件，该手机软件可以给音乐人提供一个简约平台，音乐人可以推送自己的作品，同时普通用户也可以及时获取最新的音乐专辑信息，对自己喜好的专辑进行处理，在一定程度上减小了用户获取信息的难度，方便音乐人发布自己的作品。

本应用主要以Android为技术核心，使用MySQL作为数据库，后端使用SpringBoot来提供数据接口和一般逻辑处理。系统的主要特点是能及时提供专辑信息，针对专辑发表观点。

本系统的主要功能是实现用户发现自己喜欢的专辑进行评论下载收藏的过程。主要功能的主要实现过程是音乐人上传自己的作品，包括歌曲文件，歌词文件，专辑封面，用户在发现页面浏览专辑并收听最后评价。本论文主要围绕基于Android的音乐应用程序开发的课题，主要介绍本课题的需求设计和分析以及最终达到的效果，围绕音乐应用程序的开发过程，重点说明本应用的设计背景，设计思想，设计框图和设计内容。

关键词:音乐评论应用，Android，MySQL，SpringBoot

**ABSTRACT**

Intoday'ssociety,theInternethasbecomeaninseparablepartofpeople'sdailylives,andmoreandmoreinformationisdisplayedinfrontofusthroughtheInternet.Peopleareincreasinglytrendingtowardsintegratedinformationacquisitionmethods.Forlistenersandmusicianswholikemusic,aconvenientchannelisneededtoobtainmusicinformation.Therefore,itisnecessarytodevelopamusicmobilephonesoftwarethatmeetstheneedsofcurrentusers.Themobilephonesoftwarecanprovideasimpleplatformformusicians,musicianscanpushtheirownworks,andordinaryuserscanalsogetthelatestmusicalbuminformationintime.Processingyourfavoritealbumsreducesthedifficultyforuserstoobtaininformationtoacertainextent,makingiteasierformusicianstopublishtheirworks.

ThisapplicationmainlyusesAndroidasthetechnicalcore,usesMySQLasthedatabase,andusesSpringBootasthebackendtoprovidedatainterfaceandgenerallogicprocessing.Themainfeatureofthesystemistheabilitytoprovidealbuminformationinatimelymannerandexpressopinionsonthealbum.

Themainfunctionofthissystemistorealizetheprocessthatusersfindtheirfavoritealbumsanddownloadthem.Themainrealizationprocessofthemainfunctionisthatthemusicianuploadshisownworks,includingsongfiles,lyricsfiles,albumcovers,usersbrowsethealbumonthediscoverypageandlistentothefinalevaluation.ThispapermainlyfocusesonthetopicofAndroid-basedmusicapplicationdevelopment,mainlyintroducesthedesignandanalysisoftheneedsofthetopicandthefinaleffect,focusingonthedevelopmentprocessofmusicapplications,focusingonthedesignbackground,designideas,designblockdiagramofthisapplicationanddesigncontent.

**KEYWORDS:**Musicreviewapp,Android，MySQL，SpringBoot

目录

# 第1章绪论

## 1.1课题背景

国家的快速发展和网络科技的飞速进步，给我们这个时代提供了许多挑战和机遇，与此同时人们的生活变得尤其丰富多彩，而想要在信息繁杂如瀚海的信息时代中寻找一首自己钟爱的音乐已经成为一件颇有难度的事。

网络技术的普及时代已经到来，随着手机软件技术的不断进步，特别是应用软件的广泛使用。手机的应用逐渐从以前的界面单调、出厂自带功能转向界面优美、功能实用的日常生活应用。今天的21世纪是信息时代，由于信息技术革命性的发展，社会各个领域都焕然一新，用户们普遍都在追赶时代的脚步，在这个互联网信息传播速度飞快的时代，要适应整个科技社会的变化和发展，不论是对用户，还是对所有的应用软件，这都是一个严峻的考验。随着网络技术融入人们的日常生活，人们的生活节奏变得非常之快，随之而来的就是关键信息的获取，因为在没有一个专业工具的情况下普通用户要获取自己需要的数据还是比较困难的。为了在众多数据中获取我们所需要的信息，对于音乐人和普通用户来说就需要一个专门的渠道来提供这些信息。因此，有必要开发一款符合当前用户需求的应用程序，用于提高用户信息获取效率，给音乐人提供一个展示自我的平台。

## 1.2课题研究目的及意义

起初，互联网发展没有这么普遍的时候，音乐人和普通用户之间的交流是非常少的，这既不利于音乐的进步又限制了普通用户的音乐赏析范围。随着社会的发展，科技的进步，越来越多的流行歌曲进入大众的视野，随之而来的就是信息的过度膨胀，人们无法从茫茫歌曲海中筛选自己喜欢钟爱的歌曲，所以这已经不能满足人们的快节奏的生活趋势，因此需要另一种新的技术来改善这个局面，音乐评论软件便成为最佳选择。

在这个信息量巨大的社会中，很多用户不能快捷有效的找到自己心怡喜欢或者当下流行的音乐，让本来就压力巨大的快节奏的生活缺少了一剂重要的调味品。音乐评论软件就是为了满足广大用户的需求而产生的，它的出现不仅有效的解决了用户和音乐人的沟通平台问题，也促进了我国流行音乐的发展和多元化，更是为广大用户在繁忙的工作生活中带去了精神上的放松和享受。为设计出一款符合用户需求的音乐评论软件，本文专门参考了现在市场上最流行的评论类软件和论坛，来满足广大用户的迫切需求，同时也为我国音乐艺术的发展做出自己的贡献。

## 1.3本文各章主要内容介绍

# 第2章相关技术介绍

## 2.1开发工具选择

### 2.1.1AndroidStudio工具介绍

AndroidStudio是Android的官方IDE。它专门为Android而打造，基于IntellijIDEA，可以加快开发速度。它的文件目录结构有多种显示方式，方便开发者清晰了解项目文件结构。AndroidStudio集成了大量的功能，方便开发者的使用，AndroidEmulator给开发者提供了便捷的模拟器环境，可以模拟硬件功能，比如GPS、多点触控等。

在进行XML布局时，AndroidStudio也提供了可视化的视图拖放功能，只需要将视图拖拽至目标位置，对其进行视图上的约束即可。在管理资源文件这方面，AndroidStudio也提供了专门的资源管理器VectorAssetStudio可以针对性的创建图片资源，可以使用Google提供的MaterialDesign图标，也可以自己导入文件。

### 2.1.2IntellijIDEA工具介绍

IntellijIDEA是众多Java开发者所钟爱的开发工具之一，最重要的原因自然是它集成了大量功能，可以减少开发者重复性的工作，比如代码自动补全、自动导包、错误提示等。开发者不需要再每次都去手动配置一些基础功能，只需要配置一次就可以使用，并且它还集成了一些第三方的非常有用的插件，比如Lombok、MyBatisplugin、DataBase等，大大缩短了开发者的开发时间。

## 2.2Android技术介绍

Android有丰富的功能，因此很容易与桌面操作系统混淆。Android是一个分层的环境，构建在Linux内核的基础上，它包括丰富的功能。Android应用程序是用Java编程语言编写的，它们在一个虚拟机（VM）中运行。每个Android应用程序都在DalvikVM的一个实例中运行，这个实例驻留在一个由Linux内核管理的进程中。

Android中的活动代表了用户的一个界面，下面通过图2-2-1来具体说明活动的生命周期。

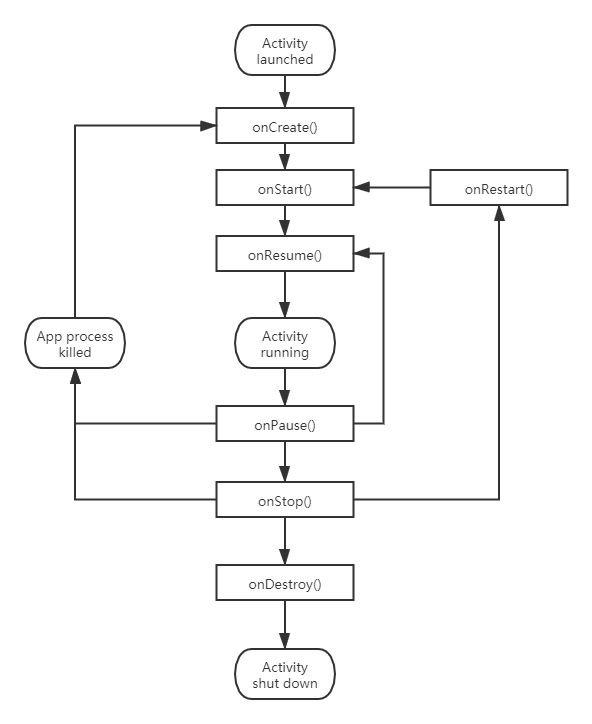


图2-2-1活动生命周期

**（1）onCreate():**这是活动第一次创建时的回调函数，可以为活动创建一个页面，在此回调中可以完成初始化操作，或者为组件绑定监听器。

**（2）onStart():**此回调在用户界面可见时调用。

**（3）onResume():**当用户和应用开始交互时，此回调函数被调用，在此状态下，此活动会被放置到堆栈的顶部。

**（4）onPause():**该方法在一个活动跳转到另一个活动的时候被调用，此时的用户可以看见页面发生变化，但无法操作，系统也会自动结束上一个活动占用的内存。

**（5）onStop():**调用此方法后，下一个活动会覆盖当前活动，用户不可见。

**（6）onDestory():**该方法发生在活动被销毁的时候，此时会自动释放所占内存，当调用**finish()**方法时也会调用此方法。

（7）**onRestart():**当活动停止或重新启动时调用。

### 2.2.1AndroidJetpack技术介绍

JetPack是一套开发库，可以帮助开发开发出优质的编写应用，其中组件可以单独使用，同时支持Kotlin语言提高工作效率，JetPack包含许多有用的组件，其中最常用的就是数据绑定组件、生命周期感知组件、LiveData类、ViewModel类等。

数据绑定，以声明的方式将可观察的数据绑定到界面元素，借助布局文件中的数据绑定组件，可以移除Activity中许多界面框架调用，使其维护起来简单、方便，并且有助于避免空指针的异常的发生。

生命周期感知型组件，可以通过感知组件的生命周期来影响另一个组件的状态变化，一般这种组件都有对应的生命周期函数，可以在对应的函数中编写代码。

LiveData是一种存储类，他的特点使具有生命周期感知能力，这种感知能力可以保证LiveData仅更新处于活跃状态的组件观察者。

ViewModel重点关注以生命周期的方式管理和存储页面数据，使其在页面反转等配置更新之后继续存在。

## 2.3开发语言介绍

在开发Android端时我采用了Kotlin语言，开发后端接口时采用了Java语言。

### 2.3.1Java语言介绍

Java是SunMicrosystems公司推出的高级程序设计语言，可以运行在多个平台，是一门面向对象的编程语言。具有简单、面向对象、分布性、编译和解释性、稳健性和安全性，内置垃圾管理器，不需要考虑内存管理，并且有最广泛的开源社区支持，各种高质量组件随时可用。

使用Java开发需要安装JDK（JavaDevelopmentKit）环境，JDK是整个Java的核心

### 2.3.2Kotlin语言介绍

Kotlin是一款开源的编程语言，可以运行在Java虚拟机上，是一种静态类型的变成语言，可以编译成Java字节码，也可以编译成JavaScript，方便运行在没有JVM（JavaVirtualMachine）的设备上，Google在2017年宣布Kotlin成为Android的官方开发语言。Kotlin的特点就是简介、安全和互操作性，它大大减少了样板代码，同样功能的代码用Kotlin编写可能会比用Java编写的代码少一半的行数，Kotlin还有一大优点就是避免空指针异常，保护开发者对空类型的误操作，由于Kotlin可以运行在JVM上，所以它可以利用JVM上任何现有的库，做到100%兼容。

## 2.4SpringBoot技术介绍

SpringBoot是在Spring的基础上通过简化配置方式来进一步简化整个应用的的搭建和开发过程。而Spring又是一个开源框架，是为了解决企业级的应用开发复杂性而创建的，为了降低Java开发的复杂性，Spring采用基于POJO的轻量级和最小侵入性编程，通过使用面接接口和依赖注入来实现松耦合，基于切面和惯例进行声明式编程，通过提供各种starter，只需要在pom文件中引入对应的start，由对应的starter指向具体的依赖、配置，从而减少spring应用配置的复杂程度，不光支持提供各种starter，还内嵌了Tomcat，不需要部署war就可以直接运行。

## 2.5MySQL技术介绍

在介绍MySQL之前需要先了解一些关于数据库的基础概念。

**DB（DataBase）：**数据库，在硬盘上以文件的形式存在。

**DBNMS（DataBaseManagementSystem）：**数据库管理系统，常见的有MySQL，Orcale等。

**SQL（StructuredQueryLanguage）：**结构化查询语言，是一种标准的语言。

**表（table）：**是数据库的基本组成单元，一个表应该包含行和列，行被称为数据记录，列被称为字段，每一个字段都应该包含相应的属性，类型和约束

**SQL语句分类：**

**（1）DQL（数据查询语言）：**查询语句，凡是select语句都是SQL

**（2）DML（数据管理语言）**：insert、delete、update对表数据进行增删改

**（3）DDL（数据定义语言）：**create、drop、alter对表结构的增删改

**（4）TCL（事务控制语言）：**commit提交事务，rollback回滚事务。

MySQL是一个关系型数据库管理系统，这种数据库将不同数据保存在不同表中，而不是保存在一个大库当中，这样可以提高数据访问速度和灵活性，MySQL支持大数据库，在多种系统上都可以运行。由于MySQL体积小、速度快、总体拥有成本低、开放源代码，一般中小型的网站开发都选择MySQL作为网站的数据库，并且支持多线程，充分利用了CPU资源，提供了TCP，JDBC等多种连接方式。

# 第3章系统分析及总体设计

## 3.1需求分析

### 3.1.1需求分析内容

随着社会快速发展和互联网行业的蓬勃发展，必须出现一种能够应对新时代人们对音乐需求的应用软件，音乐评论软件就是理想的选择。为了设计出满足大多数用户需求的音乐评论软件，本文专门研究了各种评论网站和手机端应用程序，来最大可能满足每个用户的需求。

为了实现网络便捷化的信息处理，使用了Retrofit、Glide、Tomcat、SpringBoot、InternetInformationServices(IIS)管理器及MySQL等相结合的技术，编写一个基于Android系统的音乐评论软件，系统的总体设计围绕一个以给广大用户提供一个及时快捷的音乐评论软件为主体的设计思想，规划一系列目标如下：

**（1）**利用网络的便捷，用户软件的多样性，开发音乐评论软件，方便普通用户和歌手之间交流，可以让歌手的作品更快的展示给用户，同时也可以从用户这里得到第一手的反馈信息，这也给了音乐人听取大众意见的机会，利于往后的艺术创作。

**（2）**用户可以随意浏览歌手发布的所有作品。

**（3）**用户可以登录，编辑自己的信息。

**（4）**用户可以在专辑列表收听歌曲，在播放列表下载和收藏专辑。

**（5）**用户可以查看自己的下载和收藏列表，可以查看最近常听的歌曲，以及听了多少次。

**（6）**用户可以注册成为歌手，成为歌手之后可以上传自己的作品。

### 3.2.2功能性需求分析

本软件为了满足用户的音乐收听和评论功能应该有以下几个功能：用户信息管理、专辑收藏、评论管理、专辑下载、专辑上传、歌词上传、封面上传。

用户信息管理功能：添加用户信息、更新用户信息、查询用户信息。

**（1）**专辑管理功能：添加专辑。

**（2）**评论管理功能：添加评论。

**（3）**专辑下载功能：下载歌曲。

**（4）**专辑上传功能：上传专辑。

**（5）**歌词上传功能：上传歌词。

**（6）**封面上传：上传封面。

1.通过对功能的分析后，总结出系统登录用户有两种身份：普通用户和歌手。普通用户可以浏览专辑，查看评论，编写评论，收藏歌曲，下载歌曲，收听歌曲，查看歌词，登录注册，修改个人信息。歌手在普通用户功能的基础上增加了，上传歌曲功能，上传歌词功能，上传封面功能。现画出普通用户和歌手的用例图。

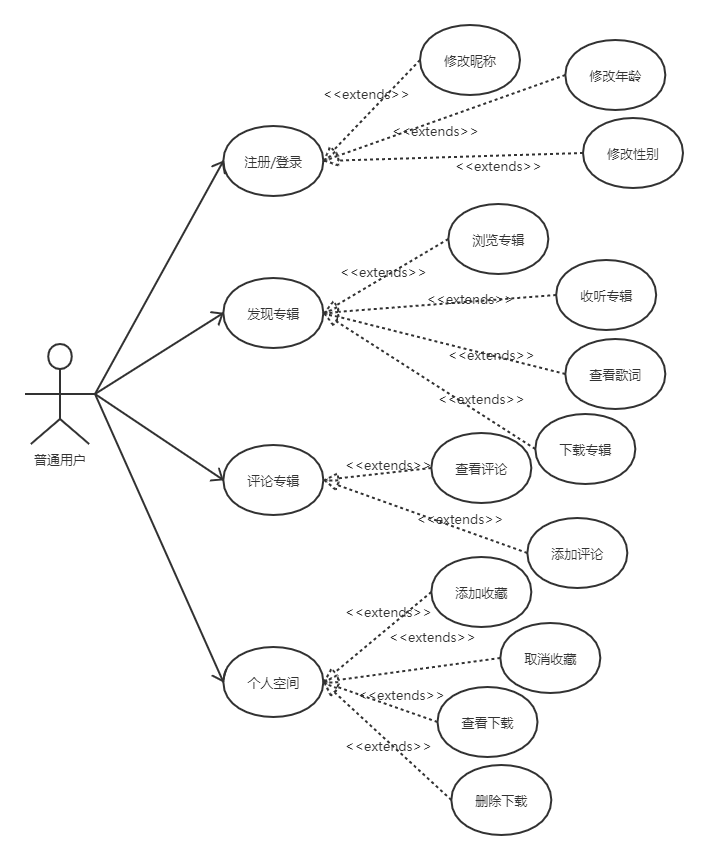


图3-1普通用户用例图

下面对这五个核心功能用例进行具体分析：

**（1）**登录/注册

在这个用例之中需要用一个登录页面来实现，当用户输入完账户和密码点击登录按钮时，系统自动去后台进行校验，如果用户名已存在则判断密码是否正确，如果密码正确则正常登录，跳转到个人信息详情页面，如果密码不正确则提示密码错误，如果用户名不存在则创建用户，在数据库中插入一条用户数据，再进入到用户详情页面。

**（2）**专辑列表

通过一个RecycleView组件来实现，列表会将从服务器取出来的数据自动适配到每一项中去，按照已经绑定好的组件位置，将数据显示到对应的组件上，每一项都可以点击跳转到对应的评论、播放页面，下载音频，查看歌词，歌词可以自动滚动，也可以手动滚动到对应的位置点击播放音频。

**（3）**专辑评论

当从查看专辑页面跳转过来之后，可以查看当前专辑下的所有评论，也可以发表自己的评论，前提是用户需要登录才可以发布自己的评论。

**（4）**个人空间

用户在登录之后可以使用这个功能，在个人空间页面可以更改用户的昵称，年龄和性别等信息，也可以选择是否成为音乐人，当成为音乐人之后，会在侧面抽屉中显示一个上传功能。

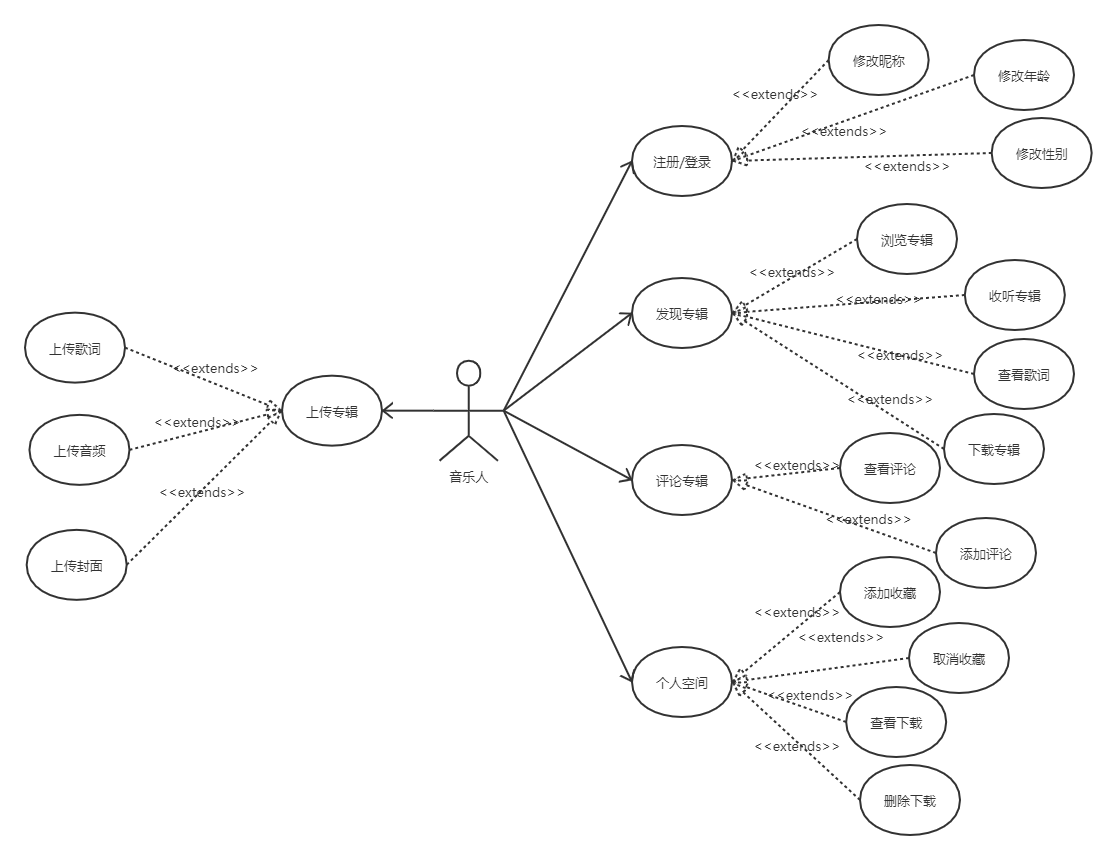


图3-2音乐人用例图

**（5）**上传专辑

当用户选择成为音乐人之后，可以上传自己的专辑，其中包括上传自己的专辑音频，专辑歌词以及专辑封面，只有这三个文件都选中之后才可以触发允许上传按钮，否则无法上传。上传的文件如果已存在会提示用户文件已存在，否则将文件上传到服务器并保存。

2.通过对画出的用例图分析，可以得出用户和音乐人的主要活动。音乐人发布自己的专辑，用户可以查看，评论，收听专辑，并且可以收藏和下载以及修改自己的信息。

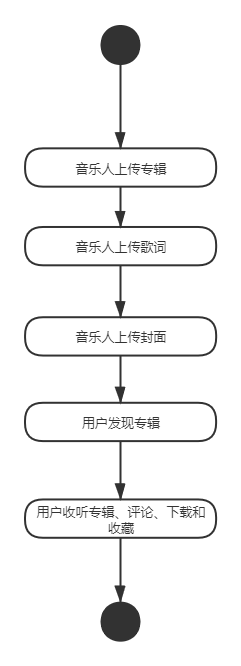


图3-3过程活动图

3.根据上述分析可以归纳出模块流程图如下所示：

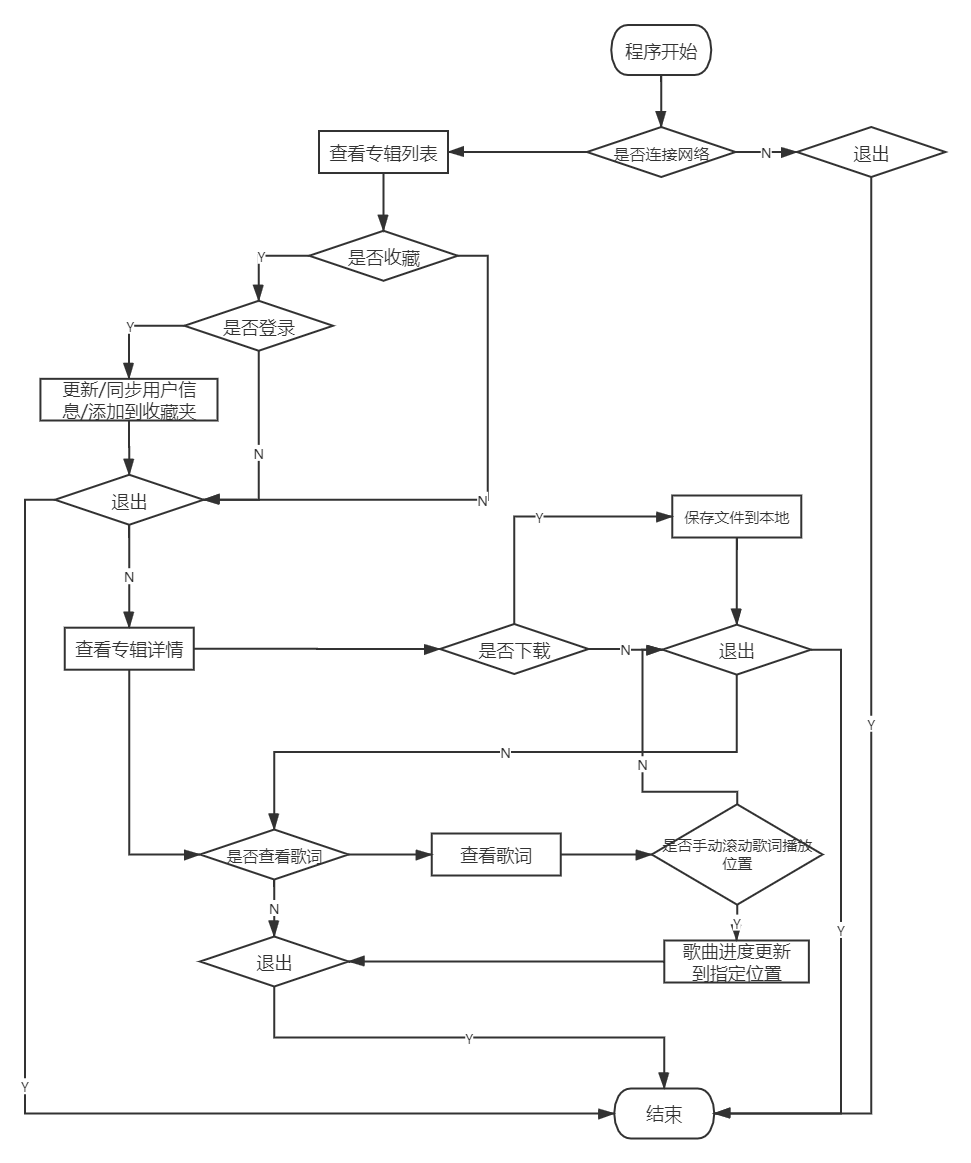


图3-4整体模块流程图

（1）打开应用软件，检查是否联网，如果没联网则结束运行。

（2）查看专辑列表，点击一个专辑可以跳转到对应的专辑评论页面，在专辑评论页面可以查看评论，下载专辑，下载的专辑可以再下载菜单中查看所有已经下载的单曲，点击专辑名称可以播放.

（3）如果需要发布专辑可以在专辑列表的侧边抽屉中点击头像登录，如果用户名存在判定密码是否正确，如果密码不正确提示密码错误，否则登录成功，进入个人空间，如果用户名不存在，则创建新的用户，跳转到用户空间.

（4）可以在个人空间修改用户信息，当登录之后再判断用户的角色是否是音乐人，如果是则显示上传功能模块，否则不显示，上传专辑时必须要将专辑封面和歌词以及专辑音频同时选中，否则无法上传，上传完成之后可以再专辑列表中去发现刚上传的专辑.

（5）查看收听记录可以再抽屉侧边栏中点击“最近常听”查看最近收听的专辑以及收听的次数，在专辑列表中可以点击右上角的收藏按钮收藏专辑，如果是已经收藏过的专辑按钮将不再显示为可点击，收藏过的专辑可以在左边抽屉栏中点击最近收听进行查看，点击任意单曲可以直接跳转到对应的播放页面。

### 3.2.3非功能性需求分析

**（1）性能需求**

**①网络环境下多用户系统。**信息主要存储在服务器端的数据库当中，根

据用户类别的不同，进行更改、添加等操作，不可以跨用户操作。

**②数据完整性和正确性。**要求插入的数据信息符合数据库的约束，限制

类型、长度以确保数据的完整性和正确性。

**③数据安全性。**根据用户类型的不同加以权限设定，不同类型的用户权

限使用范围不同，通过此法来保证用户的个人信息不外露，提高个人信息数据安全。

**（2）环境需求**

环境的配置如下表3-1所示：

表3-1环境需求配置表

|  |  |
| --- | --- |
| 系统  配置 | 系统：Windows10专业版。  显卡：Intel(R)UHDGraphics620  处理器：Intel(R)Core(TM)i5-8265UCPU@1.60GHz1.80GHz  RAM：运行内存8GB  磁盘空间：500GB |
| 软件  配置 | MySQL数据库  IntellijIDEA2019.3  AndroidStudio3  NavicatforMySQL |

# 第4章详细设计

## 4.1系统开发步骤

根据需求分析阶段的分析研究后，可以总结出，音乐评论软件的的开发过程可以分为三个阶段：前期的需求分析和大概的功能设计，中期的详细功能设计和细化功能实现，后期时维护和测试。每个阶段都以严格的线性顺序进行开发。

## 4.2功能模块设计

在需求分析阶段，首先需要想清楚这个软件是为了解决什么问题而出现的，然后围绕这个目标展开一系列的分析，形成逻辑模型，再根据形成的逻辑模型转换为我们自己的物理模型，根据需求的功能划分独立的模块，形成模型图，最后根据模型图实现具体的需求，以及美化界面。根据软件的需求大致划分了几个模块：登录模块，用户信息管理模块，专辑列表模块，播放/评论模块，下载/上传模块，收藏模块，如图4-2所示。

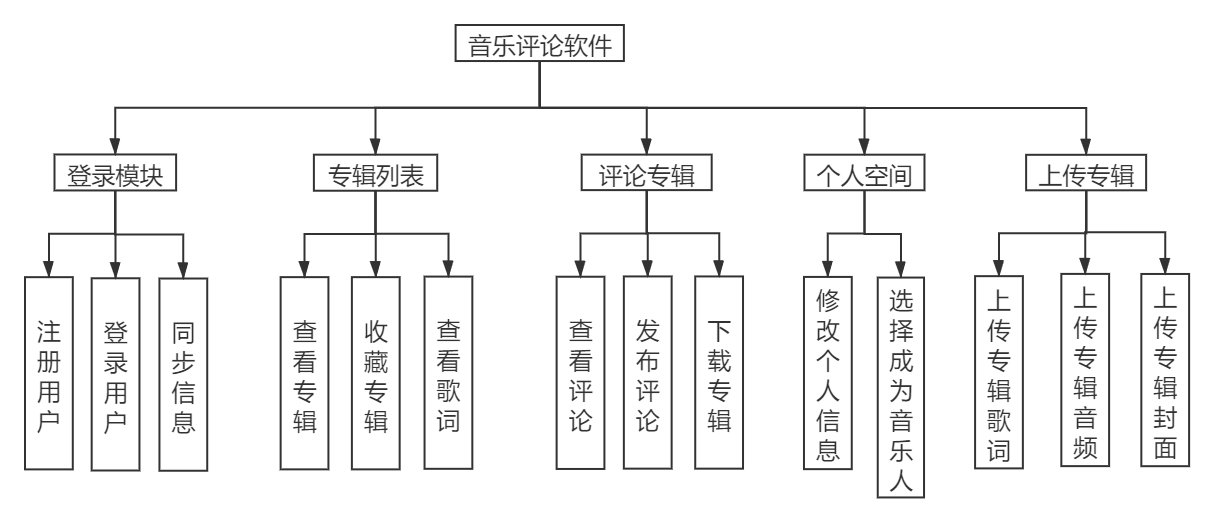


图4-2软件架构图

1. 登录模块

登录模块界面包括：用户的登录，用户的注册以及后台的用户数据同步，该界面是用户需要记录或者同步自己的历史记录时所必须显示的页面。

（1）注册用户

通过一个按钮实现用户的注册和登录，两者的区别在于：如果用户名正确，则判断密码是否正确，如果密码正确则进入下一个页面，否则提示密码错误，如果用户名不存在则认为是创建新用户，在数据库中插入一条数据，这样通过一个按钮解决了登录和注册两个功能，提高了按钮控件利用率，使空间更为简洁大方。

（2）登录用户

和注册用同一个按钮，通过判断用户名进行下一步操作，登录成功之后会跳转到用户详情页面。

（3）同步信息

这个功能是在用户登录成功之后后台会自动触发这个功能，利用网络和数据库，将用户之前的历史记录都同步到本地，例如收藏列表，最近常听列表都将会持久化到本地文件保存，这样设计会增加用户好感度、舒适性，让用户体验得到提高。

1. 专辑列表

专辑列表模块包括，查看专辑，收藏专辑，查看歌词等功能，该界面是用户的主要界面，用户第一次打开软件从开始页面跳转过来的就是专辑查看列表页面。

（1）查看专辑

查看专辑功能是主要功能之一，主要是通过网络请求，将专辑的数据通过适配器加载到列表中，可以向下滑动，适配器会自动加载即将进入页面的数据将其显示出来，点击每一专辑可以进入每一个专辑对应的评论、播放页面。

（2）收藏专辑

在专辑列表中每一个专辑面板上都有显示一个收藏按钮，在用户已经登录的情况下会自动辨别该专辑是否已经被该用户收藏，如果已经被用户收藏该按钮将会显示为不可被选中，并显示“已收藏”字样。当点击收藏按钮时将会把该专辑加入到用户的收藏夹中，可以在用户的收藏夹中查看收藏的专辑列表。

（3）查看歌词

当从一个专辑中进入之后，可以看到该专辑的评论以及播放按钮，还有歌词按钮以及下载按钮。点击查看歌词按钮可以查看专辑对应的歌词，滑动歌词可以改变歌曲的播放进度。

1. 评论专辑

评论专辑模块主要包括查看评论、发布评论、下载专辑等功能。评论专辑的界面属于音乐评论软件的核心部分，在评论模块用户可以和音乐人进行交流，这也是一个两种角色可以交流的一个模块，可以帮助音乐人看到更多的想法，有助于艺术的进步，也给用户提供了一个可以发声的平台。

（1）查看评论

当进入到查看专辑页面时，通过网络请求获取该专辑下的所有评论，再通过RecycleView加载到每一项。

（2）发布评论

给用户提供一个发表自己想法的窗口，窗口没有设置字数限制，可以发布大段的文章，发布窗口中有输入的时候，发布按钮才会显示为可以点击，否则显示为无法触发。

（3）下载专辑

下载专辑是通过评论页面的一个下载按钮来实现的，通过资源服务器上的资源连接下载到本地保存，下载的只有专辑音频，可以再下载菜单中查看所下载的专辑，点击可以直接播放。

1. 个人空间

个人空间是只有在用户成功登录之后，才会跳转的页面，通过在这个页面上操作对应的空间可以更改用户的对应信息，例如点击“昵称”、“性别”，可以直接更改用户的信息，更改之后可以直接保存到数据库，同时将用户信息更新到本地存储文件中。其中可以更改用户角色，是否成为音乐人，点击按钮可以看到相关的条款、规范和约束，同意即可成为音乐人，可以开启上传专辑功能。

1. 上传专辑

上传专辑包括三个必不可少的功能，即上传歌词文件、上传专辑音频和上传封面。当三个文件都选中的时候才可以使上传按钮变成可点击的，文件上传到服务器之后。根据文件后缀将文件存储在服务器不同的文件中。

（1）上传歌词文件

上传歌词文件按钮对歌词格式进行了约束，只有lrc格式的文件可以选择。

（2）上传专辑文件

上传专辑对音频文件进行了约束，音频格式为MP3的文件可以上传。

（3）上传专辑封面

选择图片文件进行上传。

## 4.3数据库设计

数据库的设计是软件设计中一个重要的部分，数据库设计主要可以从两方面研究。分别是数据库存储设计以及数据库字段设计。

1、数据库存储设计

现在有很多数据库，例如非关系型数据库Redis、MongoDB等，关系型数据库MySQL、Oracle等。本文主要以MySQL数据库作为存储。MySQL是由瑞典MySQLAB公司开发，属于Oracle公司旗下的产品，它是将数据保存在一张张表中，而不是整个大的仓库中，如图是数据库设计ER图。

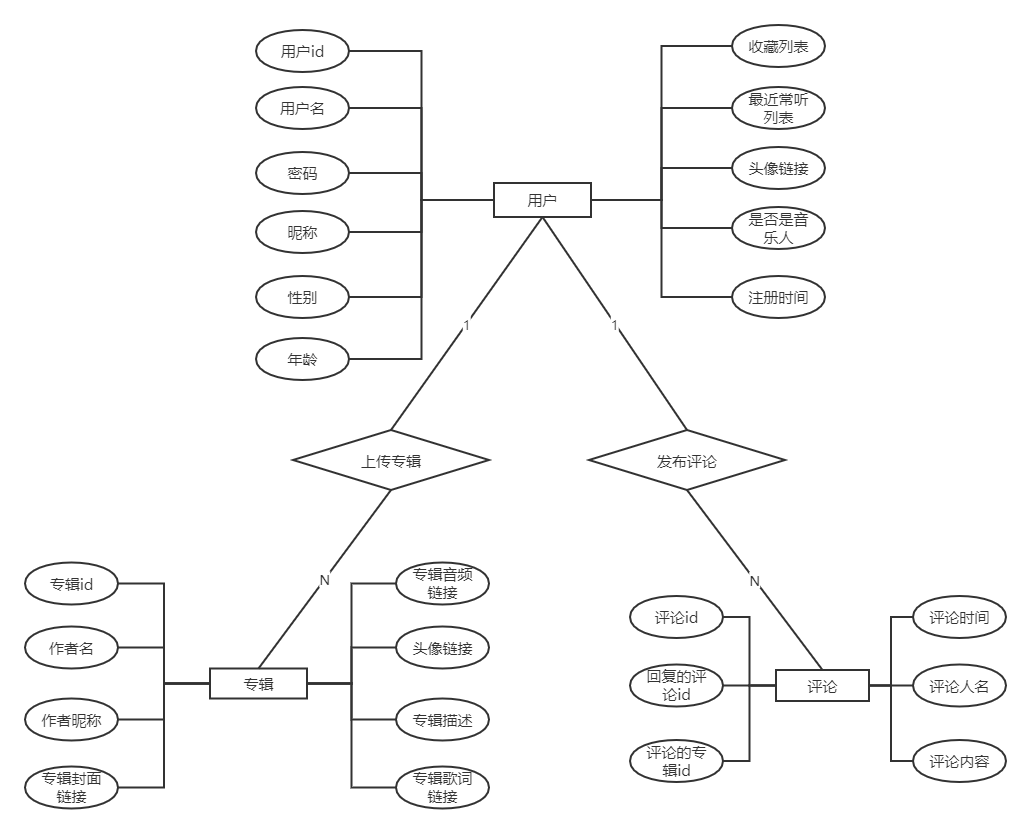


图4-3数据库设计ER图

2、数据库字段设计

根据音乐评论软件的主要功能，数据库设计字段主要研究了三个表的字段设计：用户表、专辑表以及评论表，下面将数据库的主要字段进行详细设计。

（1）用户表，该表主要存放用户详细信息，主要包含用户id、用户名、密码、昵称、性别、年龄、收藏列表、最近常听、头像链接、是否是音乐人、注册时间，如表4-4所示：

表4-4用户表字段

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 字段类型 | 说明 | 主键 | 空值 |
| **id** | **INT** | 用户id | **PrimaryKey** | 不允许 |
| **name** | **VARCHAR** | 用户名 |  | 不允许 |
| **password** | **VARCHAR** | 密码 |  | 不允许 |
| **nick\_name** | **VARCHAR** | 昵称 |  | 允许 |
| **sex** | **CHAR** | 性别 |  | 允许 |
| **age** | **VARCHAR** | 年龄 |  | 允许 |
| **register\_date** | **DATETIME** | 注册日期 |  | 允许 |
| **collect\_list** | **TEXT** | 收藏列表 |  | 允许 |
| **recently\_listen** | **TEXT** | 最近常听记录 |  | 允许 |
| **is\_singer** | **INT** | 是否是歌手 |  | 允许 |
| **haeder\_image\_url** | **VARCHAR** | 头像链接 |  | 允许 |
| **upload\_date** | **VARCHAR** | 上传日期 |  | 允许 |

（2）专辑表，该表主要存放所有的专辑信息，包括专辑id、创作人名称、创作人昵称、专辑封面链接、专辑歌词链接、专辑音频链接、专辑名称、头像链接、专辑描述等字段，如表4-5所示：

表4-5专辑表字段

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 字段类型 | 说明 | 主键 | 空值 |
| **id** | **INT** | 专辑id | **PrimaryKey** | 不允许 |
| **producer\_name** | **VARCHAR** | 制作人名称 |  | 允许 |
| **producer\_nickname** | **VARCHAR** | 制作人昵称 |  | 允许 |
| **album\_cover\_url** | **VARCHAR** | 专辑封面 |  | 允许 |
| **media\_lrc\_url** | **VARCHAR** | 专辑歌词链接 |  | 允许 |
| **media\_url** | **VARCHAR** | 专辑音频链接 |  | 允许 |
| **album\_name** | **VARCHAR** | 专辑名称 |  | 允许 |
| **header\_icon** | **VARCHAR** | 头像链接 |  | 允许 |
| **content\_desc** | **VARCHAR** | 专辑描述 |  | 允许 |

（3）评论表，该表主要包括评论id、回复的专辑id、回复的评论id、评论时间、评论人名称、评论内容等字段，如下表4-6所示：

表4-6评论表字段

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 字段类型 | 说明 | 主键 | 空值 |
| **id** | **INT** | 评论id | **PrimaryKey** | 不允许 |
| **reply\_album\_id** | **INT** | 评论的专辑id |  | 允许 |
| **reply\_comm\_id** | **INT** | 回复的评论的id |  | 允许 |
| **comm\_time** | **DATETIME** | 评论时间 |  | 不允许 |
| **comm\_name** | **VARCHAR** | 评论人名称 |  | 允许 |
| **comm\_content** | **TEXT** | 评论内容 |  | 允许 |

三个表之间的关系是相互独立的，没有外键关联，关系如下图4-7所示：

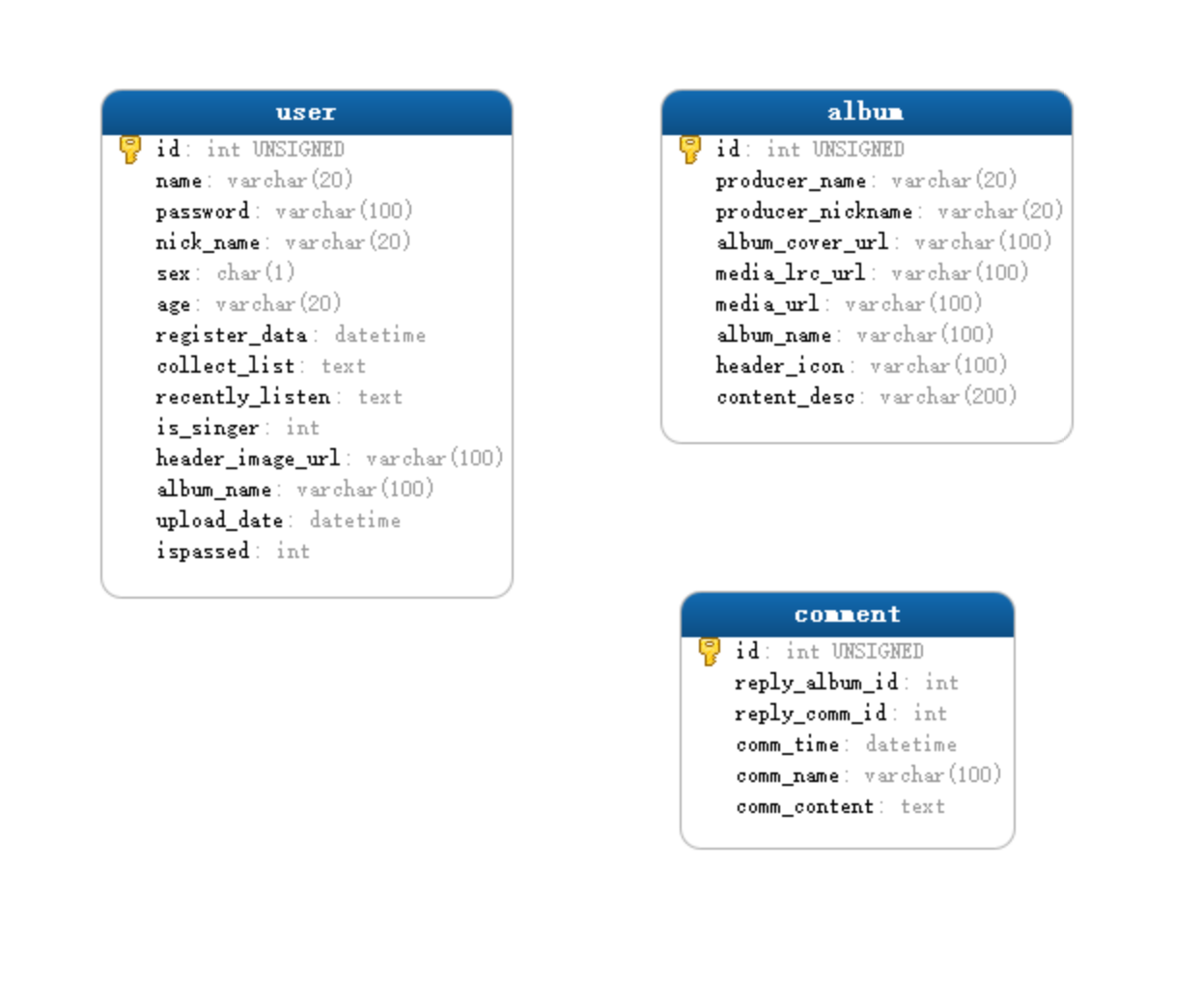


图4-7三表之间的关系图

## 4.3平台播放界面设计

### 4.3.1Android界面层级结构及常用布局介绍

现在的软件发展越来越迅速，不仅功能越来越符合人们的需求，而且界面也随之变得更加简约时尚。在软件研究中如何实现一个满足用户个性化的界面成为了一项重要内容，在Android中布局的所有元素均使用View和ViewGroup对象的层次结构进行构建，View可以绘制用户可见的交互内容，ViewGroup是容纳View和其他ViewGroup的容器，不可见。视图结构层次如图4-8所示：

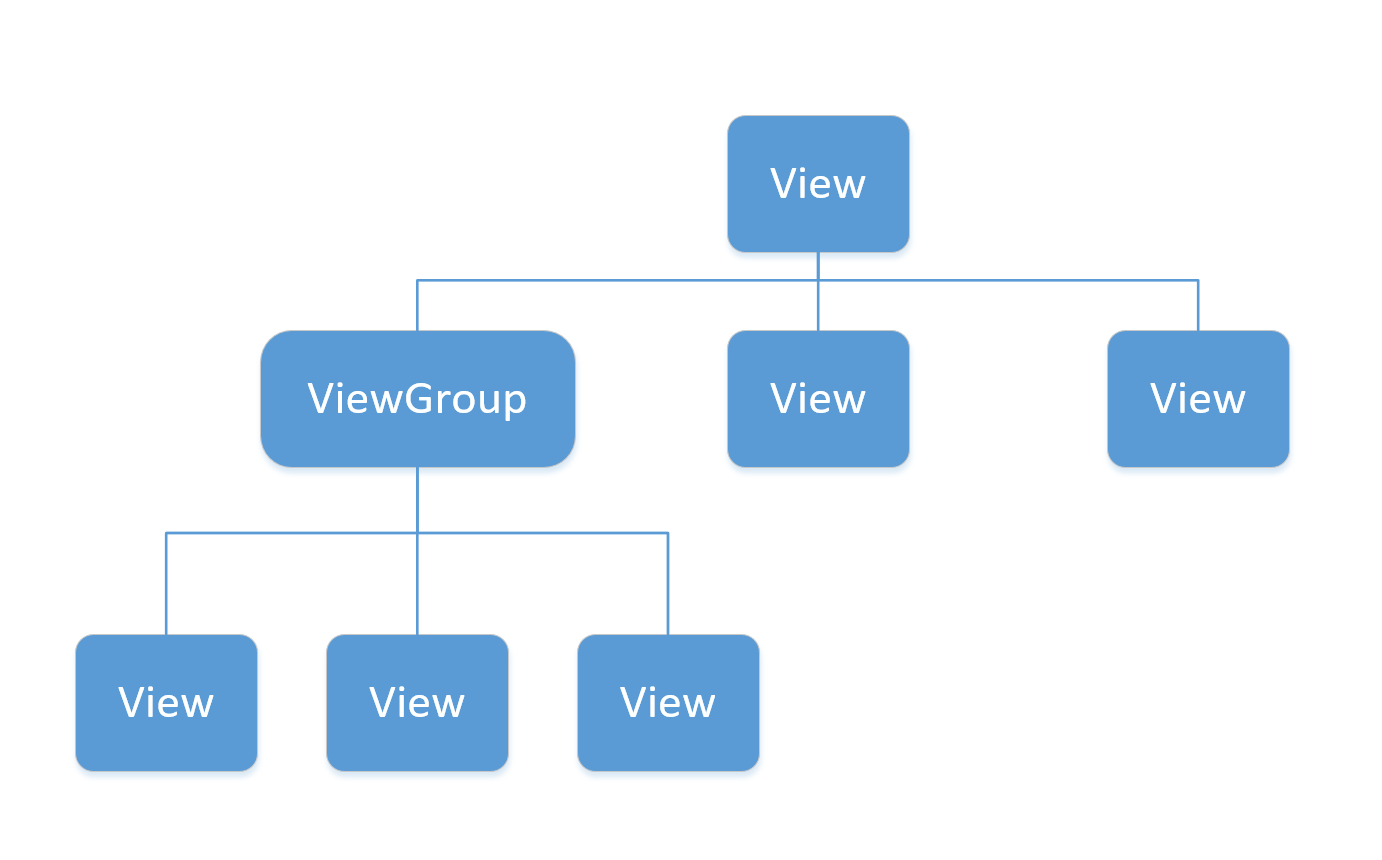


图4-8视图层级结构图

将控价摆放在容器之中进行操作，其中主要由以下几种布局控件：

1、线性布局（LinearLayout）

线性布局是布局中最常用的布局方式，可以通过“android:orientation”来指定控件摆放方向，vertical和horizontal分别指定方向为垂直和水平，默认是水平方向。通过android:gravity来确定内部控件对齐方式，常用属性值有center、center\_vertical、center\_horizontal、top、bottom、left、right等。

2、相对布局（RelativeLayout）

可以让子控件相对于父控件或者兄弟控件进行位置摆放。可以实现多重嵌套效果，需要注意的是相对布局一定要加id才可以使用。

3、帧布局（FrameLayout）

帧布局是按照从屏幕左上角按照不同层次堆叠方式布局，后面的可以覆盖前面的控件，当需要实现层面显示的时候可以使用。

4、绝对布局（AbsoluteLayout）

绝对布局是将控件以坐标的形式固定下来，如果使用绝对布局，可能导致在所有终端上显示不全等问题。

5、网格布局（GridLayout）

和网格布局一样，可以设置显示的方向，容器中的组件可以跨越多行也可以跨越多列显示。

### 4.3.2系统主要界面

由于本文主要研究的应用软件主要包括以下几个功能，每一个都会有一个主界面，在这个章节中主要研究专辑查看界面、评论/播放界面、侧边抽屉界面、查看歌词界面、用户登录界面、用户详情界面、收藏列表界面、下载列表界面。

1、查看专辑界面布局

查看专辑页面布局的研究主要是考虑用户需要与界面交互，因此设计一个页面简洁而不失简单的界面尤为重要，故通过学习布局相关知识设计出如下如图4-9的具体模型。



图4-9查看专辑界面模型

查看专辑界面使用了DrawerLayout布局控件来实现，DrawerLayout分为侧边菜单和主内容区两部分，侧边菜单可以根据手势展开与隐藏，主内容区的部分可以随着菜单的点击而变化。DrawerLayout和LinearLayout是同一种控件，只是比LinearLayout多一个滑动功能。DrawerLayout的第一个子元素是主要内容，即抽屉没有打开时显示的布局。DrawerLayout的第二个子元素是抽屉中的内容，即抽屉布局，在后面介绍抽屉布局。主体布局是通过嵌套多个LinearLayout来实现的，在LinearLayout中加入RecycleView来实现列表的滑动加载。

图中数字代表的具体意思如下：

1. 显示用户头像，用圆形的ImageView插件添加
2. 显示用户昵称
3. 显示收藏按钮，当用户已经收藏了该专辑时，显示效果为不可点击
4. 显示专辑介绍，以文本的形式
5. 展示专辑封面
6. 显示专辑名称
7. 侧边抽屉

2、评论/播放界面

播放/评论页面显示的是用户的评论以及查看歌词、播放专辑、下载按钮，页面下方有评论框和发送评论按钮，是通过折叠菜单栏和滑动列表实现的，主要布局代码如下所示：

|  |
| --- |
| <androidx.coordinatorlayout.widget.CoordinatorLayout>  /…/  <com.google.android.material.appbar.AppBarLayout>  <com.google.android.material.appbar.CollapsingToolbarLayout>  /…/  <androidx.appcompat.widget.Toolbar  /…/  />  </com.google.android.material.appbar.CollapsingToolbarLayout>  </com.google.android.material.appbar.AppBarLayout>  <androidx.core.widget.NestedScrollView  /…/>  <LinearLayout/…/>  <androidx.recyclerview.widget.RecyclerView/>  </LinearLayout>  </androidx.core.widget.NestedScrollView>  </androidx.coordinatorlayout.widget.CoordinatorLayout> |

这里的页面开发用到了CoordinatorLayout，CoordinatorLayout的作用是作为顶层布局，可以协调子View之间交互的容器，使用CoordinatorLayout需要在build.gradle加入：**implementation'com.android.support:design:25.1.0'。**AppBarLayout是一个垂直布局的LinearLayout，它主要是为了实现**“MaterialDesign”**风格的标题栏的特性，例如滚动。CollapsingToolbarLayout继承自FrameLayout，它是用来实现Toolbar的折叠效果，一般它的直接子View是Toolbar，用来实现菜单折叠效果。RecycleView的实现显示过程如下图4-10所示：



图4-10RecycleView的信息加载显示过程

3、侧边抽屉界面

侧边抽屉栏可以直接通过手指滑动打开，其中可以显示，用户的收藏夹、头像、下载列表、最近收听列表以及上传按钮，其中上传按钮只有用户在注册成为音乐人角色之后才可以使用，否则是不会显示给普通用户的。其侧边模型如下图4-11所示:



图4-11侧边抽屉模型图

图中数字及按钮代表的具体意思如下：

（1）头像，可以显示用户头像，点击可以进入到用户登录页面，如果已经登录将会进入用户详情页面。

收藏夹：保存用户的收藏记录，点击可以进入到用户的收藏列表中查看收藏记录。

最近常听：将用户最近听的歌曲进行统计，以列表的形式显示歌曲名和听歌次数。

我的下载：以列表的形式显示已经下载好的歌曲，在下载列表页面可以点击直接播放音乐，也可以点击每一个列表后面的删除按钮进行删除操作。

上传：这个功能只有在用户开通音乐人角色的时候才可以点亮，点击可以进入到上传页面进行一系列的上传操作。

侧边滑动栏中的核心代码如下所示：

|  |
| --- |
| <com.google.android.material.navigation.NavigationView  android:id="@+id/nav\_view"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_gravity="start"  app:menu="@menu/nav\_menu">  </com.google.android.material.navigation.NavigationView> |

|  |
| --- |
| <menuxmlns:android="http://schemas.android.com/apk/res/android">  <groupandroid:checkableBehavior="single">  <itemandroid:id="@+id/nav\_collect"  android:icon="@drawable/ic\_folder"  android:title="收藏夹"></item>  <itemandroid:id="@+id/nav\_recently"  android:icon="@drawable/ic\_queue\_music"  android:title="最近常听"></item>  <itemandroid:id="@+id/nav\_download"  android:icon="@drawable/download"  android:title="我的下载"></item>  <itemandroid:id="@+id/nav\_upload"  android:icon="@drawable/upload"  android:title="上传"></item>  <item  </group>  </menu> |

在侧边抽屉栏中使用了Menu菜单布局和NavigationView，要使用NavigationView必须让Activity主题先设置

|  |
| --- |
| <stylename="AppTheme.NoActionBar">  <itemname="windowActionBar">false</item>  <itemname="windowNoTitle">true</item>  </style> |

4、查看歌词界面

歌词界面主要通过调用第三方控件来实现歌词的加载显示和滚动，界面主要代码如下：

|  |
| --- |
| <com.lauzy.freedom.library.LrcView  android:id="@+id/mLrcView"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  app:currentIndicateLrcColor="@color/colorAccent"  app:indicatorTextColor="@android:color/holo\_orange\_dark"  app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"  app:playIcon="@drawable/ic\_play\_circle">  </com.lauzy.freedom.library.LrcView> |

通过使用开源的歌词框架LrcView来实现歌词的显示滚动功能，要使用此框架还需要再Gradle中加入implementation'com.github.Lauzy:LyricView:1.0.4'来实现功能的引用，

逻辑代码主要如下：

|  |
| --- |
| //设置歌词  mLrcView.setLrcData(lrcData)  //设置同步歌词和音频进度  mLrcView.updateTime(mediaPlayer.currentPosition.toLong())  //设置歌词按进度播放按钮监听器  mLrcView.setOnPlayIndicatorLineListener { time, content -> mediaPlayer.seekTo(time.toInt()) } |

5、用户登录界面

通过LinearLayout设置两个输入框和一个确认按钮即可实现登录页面的构建。

6、用户详情界面

通过使用CoordinatorLayout、AppBarLayout、Toolbar以及嵌套LinearLayout来实现，文字通过TextView来实现展示，设置监听器将每次更改后的信息同步到服务器。

7、收藏列表、下载列表界面

这个两个页面都是相同的，都是通过RecycleView来实现页面的滚动加载显示，不同的是收藏列表中后面的每一项按钮是收藏和取消收藏功能，下载列表后面的按钮是删除功能。

到这部分基本上所有的页面设计就完成了，当点击每一个按钮都会由相应的Activity会唤醒加载显示，Activity和Activity之间跳转传递信息通过Intent，Intent可以带参数传过去，用StartActivity可以开启另一个Activity。

# 第5章音乐评论应用程序的实现

## 5.1专辑列表功能实现

## 5.2评论列表功能实现

## 5.3专辑下载/上传功能实现

## 5.4专辑播放功能实现

## 5.5专辑收藏功能实现

# 第6章测试

# 第7章结束语