### **团队计划制定 & 需求分析与设计**

一、确立团队任务计划

根据发布的项目时间规划，初步确立团队任务计划，将团队的任务计划添加到Github的团队项目Issues（列表、里程碑）里面，要求：

（1）PM组织成员快速阅览《构建之法》第8章到第15章关于“软件开发完整生命周期”的相关章节，并回顾在案例分析中初步练习过的项目计划设计的方法对项目做时间安排。同时请参考下面的团队项目每周环节进度表。

（2）使用《构建之法》“计划和估计” 一节中提到的讨论方法和耗时估计公式来矫正计划的时间表，为每个环节定制好时间。并注意，不要贪图做太多的功能，而是要把关键的场景实现了（参考《构建之法》中关于MVP的描述）。

二、参考

### 1. 如何使用Issue管理软件项目？http://www.ruanyifeng.com/blog/2017/08/issue.html

2. GitHub团队项目合作例子：<http://www.cnblogs.com/schaepher/p/4933873.html>

三、软件需求分析报告要求

1） 参考《软件需求规格说明书》国标规范文本，撰写对应项目的软件需求规格说明书。

2） 除形式上满足规范文本要求外，整体内容必须围绕项目实质展开，对所要开发的项目确保尽力做到清晰完整准确。

3） 采用分层形式描述，随着“层”的深入，描述的内容细节越具体。

4） 使用一致的图形符号和文字描述内容。

5） 所有的缩写须事先定义。

6） 格式：需要有目录，word排版样式规范美观，图文并茂，通篇文档有一个统一的样式风格，注意字体和行距等。

7） 将自己置于读者的立场——如果对软件项目不熟悉的人员，通过阅读这份文档，能否完全读懂软件要做什么。

8） 访问软件项目的真实用户，确保软件真正体现用户的需求，为软件最终可用奠定基础。

四、参考资料

1. 用户调研方法参考《构建之法》第8章获取用户需求——用户调研 http://www.cnblogs.com/xinz/archive/2013/02/03/2890786.html

http://www.cnblogs.com/xinz/p/3308608.html

1. 分析和设计方法：http://www.cnblogs.com/xinz/p/4525232.html

在线作图工具ProcessOn：https://www.processon.com/

原型设计教程

https://www.cnblogs.com/SivilTaram/p/9190373.html

1. 原型工具参考
   1. 移动应用原型与线框工具-墨刀
   2. 原型设计界的PS -Axure RP，Axure
   3. 网页和移动端的设计sketch
   4. 一款简洁高效的原型图设计工具mockplus
   5. 致力于高保真原型制作工具Justinmind
   6. 一款免费的带有手绘涂鸦风格的原型设计软件balsamiq mockups
   7. 更多选择，请参考：https://www.zhihu.com/question/19592829

五、系统设计

(1)在设计阶段，我们要清楚：软件是怎么解决这些需求的？

(2)一个好的分层式结构，可以使得开发人员的分工更加明确。一旦定义好各层次之间的接口，负责不同逻辑设计的开发人员就可以分散关注，齐头并进。

(3)如何才能最大限度地实现这些需求，这就是架构设计要解决的问题。请给出系统的架构设计

### **练习要求**

1. **要求一 制定团队计划（实验报告以团队为单位提交）；**
   1. 6-8周：用户调研，跟用户一起确定要解决的问题和项目需求，编写需求规格说明书。
   2. 8-9周：根据需求规格说明书，将整个系统根据功能分解成若干个模块，确定每个模块之间的联系，确定每个模块需要实现的具体功能，对数据库进行设计。
   3. 8-9周：项目负责人决定需要采用什么样的技术和什么框架开发项目，将项目的基本框架搭建出来。
   4. 9-14周：开发人员根据具体的功能模块编写代码。团队项目Alpha、Beta阶段。
   5. 15周：测试。测试的目的是为了发现程序的一些不足之处。关键在于设计好的测试用例，好的用例才能发现出项目的不足
   6. 后期维护，对出现的一些bug及时进行修改，根据用户新的需求对不断对项目进行完善
2. **根据 要求二 参考1，设立里程碑（“MileStone”），学习参考2并掌握如何利用github进行团队项目合作；**

**里程碑：issue的容器，相关的issue可以放到同一个MileStone里：**

1. 制定团队计划，编写需求规格说明书，设置编码规范
2. 确定项目体系结构，设计具体功能模块，设计数据库，项目的基本框架搭建
3. Alpha冲刺阶段。每个成员的预期目标，项目的总预期目标，每个成员Alpha阶段的总结，项目Alpha界面总结。
4. Beta冲刺阶段。团队Beta阶段个人总结，团队Beta阶段报告。根据新出现的需求对项目进行修改。根据Alpha阶段出现不足对项目进行改进。
5. 项目测试。设计好的测试用例进行测试，对测试出现的bug进行修改。

**利用github进行团队项目合作：**

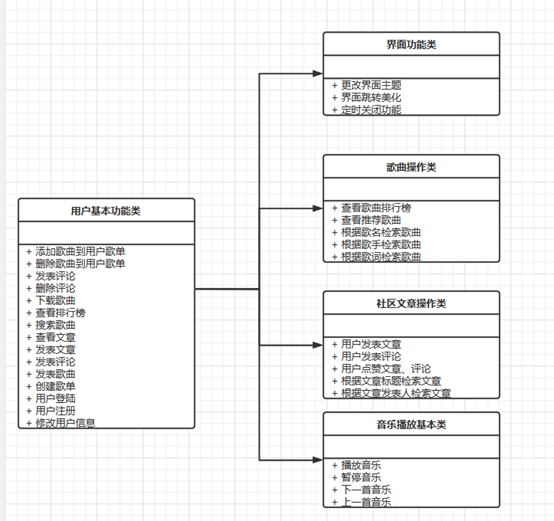
1. 创建一个团队，并邀请团队成员加入。
2. 在github上创建好仓库，用这个仓库存放项目代码，并邀请其他团队成员加入仓库。
3. 创建开发分支，master分支是项目的稳定版本，开发分支是项目的开发版本，提交代码应先提交到开发分支。
4. 使用fork，为了让队员之间不相互影响，可以让队友将仓库fork成自己的仓库，然后分别进行开发。
5. 将github上的云端仓库克隆到本地。
6. 本地仓库与远程仓库同步。从上游仓库fetch分支拉取数据，并提交给本地master，会存储在upstream/master上，切换到本地主分支，将upstream/master分支上合并到本地master上，这样就会完成了同步，并且本地内容也不会丢失，将本地变动提交到Github的fork。
7. 使用push将本地的修改同步到远程的github仓库上。
8. 将修改反馈到团队仓库。如果fork下来的项目领先或者落后上游分支，可以在网页上进行new pull request，并等待负责人审核是否同意采纳你的修改。
9. **给出本团队项目的目标用户（如果有必要，请用问卷调查等方式）**

音乐爱好者

1. **根据 要求三 完成需求分析报告**
   1. 引言 、
2. 编写目的

为了明确项目需求，规划好项目进度，组织各成员之间开发与测试，撰写此文档。

1. 项目背景
   1. 待开发软件名称：网乐云（暂定）
   2. 项目任务提出者：严威
   3. 项目开发者：该队不会UI队各成员
   4. 本说明书作者：该队不会UI队各成员
2. 参考资料
   1. 《阿里巴巴Java开发手册》
   2. 《软件需求规格说明书》国标规范文本
   3. 《构建之法》作者：邹欣
   4. 《如何使用Issue管理软件项目》<http://www.ruanyifeng.com/blog/2017/08/issue.html>
   5. 《GitHub团队项目合作流程》<http://www.cnblogs.com/schaepher/p/4933873.html>
   6. 用户场景、
3. 使用时间：午休时光；睡觉前；等车的空档期；写作业时；运动中；认真工作时；如果你有时间，甚至24小时你都可以使用。
4. 使用地点：吵杂的车厢；自习室里；工作的电脑前的你；走在路上的你（注意看路，耳机音量过大可能让你听不到车的喇叭声）；健身房里； 还能你能想象到的任何地方。
5. 使用目的：通过音乐找到共鸣；听舒缓的音乐催眠；在音乐中寻找安慰，获取快乐；找到志同道合的音友；分享一则关联音乐的说说；发表自己的新曲，等等
   1. 类图、



* 1. 界面原型

进入app： 我的：

点击账号： 播放界面：

歌单： 社区：

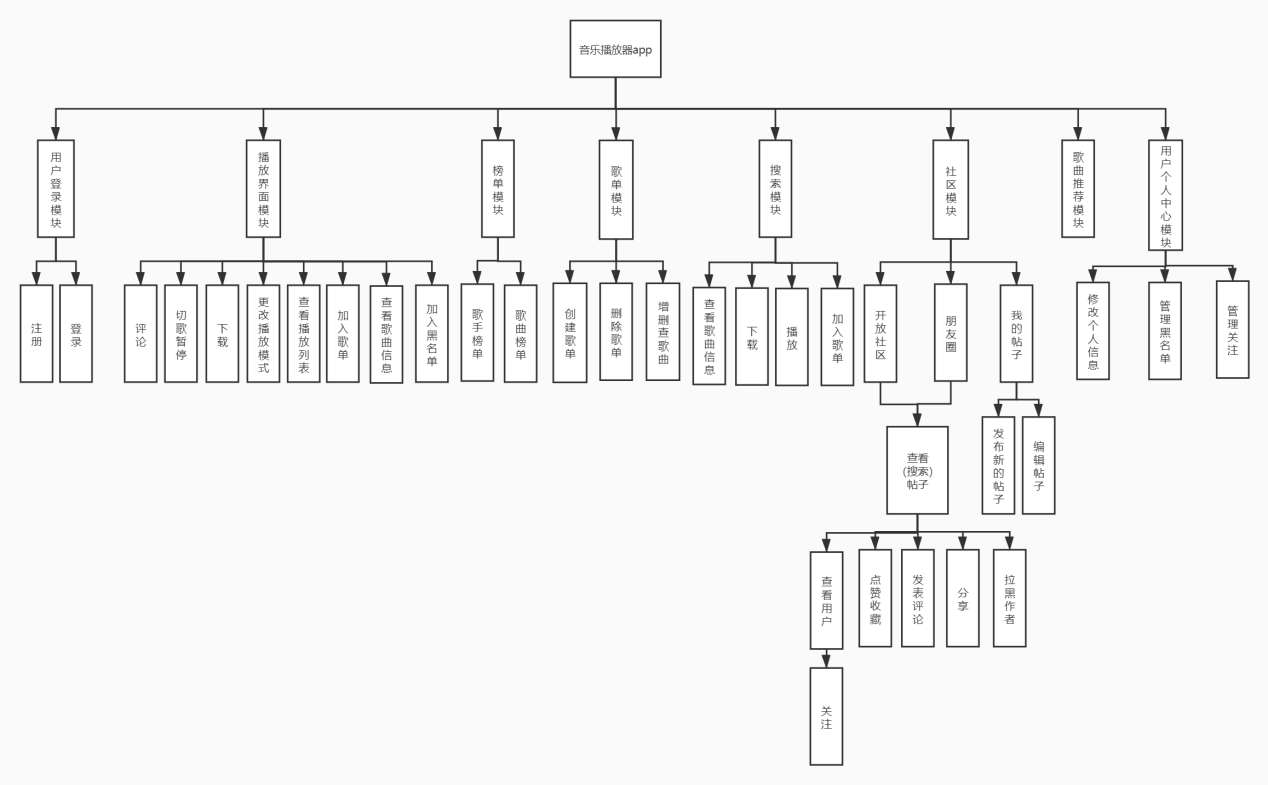
 

* 1. 功能描述、

1. 播放器基本功能：播放音乐、播放上一首音乐、播放下一首音乐、暂停、查看播放列表、调整播放音乐的音量、后台播放音乐、查看歌曲的基本信息。
2. 对个人的歌单管理：添加歌曲到个人歌单、从个人歌单中删除歌曲。
3. 搜索功能：根据歌名、歌词和歌手模糊搜索。
4. 界面：界面简洁、用户友好界面、可以设置界面主题。
5. 社区：一个音乐属性的社区，用户可以在社区相互交流。
   1. 验收验证标准、
6. 项目应实现需求中的播放器基本功能，并且不存在严重bug。
7. 代码应符合编码规范
8. github上应有每个成员的项目开发过程，每个人都有对应的工作
9. 准确且细致的文档
10. 全面且合理的数据库 设计
11. 设计测试用例，并且项目能通过测试。
    1. 文档的图表、文字、样式统一且符合规范
12. 将需求分析报告上传至团队的github上
13. **功能分析和任务分解WBS**
    1. 参考《构建之法》P173图8-8，给出功能分析的四个象限定位

|  |  |  |
| --- | --- | --- |
|  | 外围功能 | 杀手功能 |
| 必要需求 | 1.用户登录注册功能  2.播放歌曲功能  3播放音乐时的切换，暂停  4.播放顺序切换(随机、顺序等)  5.歌曲分类展示(按歌手、类型)  6.歌曲的下载和删除  7.歌曲的检索功能  8.歌曲的收藏功能  9.歌曲推荐功能  10.用户中心功能 | 18.社区发表个性化帖子  19.修改已发表帖子  20.社区帖子评论功能  21.社区帖子点赞功能  22.朋友圈功能 |
| 辅助需求 | 11.歌单功能  12.歌曲评论功能  13.排行榜功能  14.页面跳转美化  15.装扮功能  16.定时关闭功能  17.音乐(歌手)黑名单 | 23.帖子筛选功能  24.帖子权限设置功能  25.界面简洁  26.帖子黑名单 |

* 1. 请给出团队项目的WBS；



* 1. 团队成员估计各自任务所需时间

严威：播放界面模块，社区模块（开放社区）

所需时间：3周

曾俊伟：播放界面模块，社区模块（朋友圈）

所需时间：3周

陈梓灿：社区模块（我的帖子），歌曲推荐模块

所需时间：3周

王鑫杰：榜单模块，搜索模块

所需时间：3周

林炜：登录模块，个人中心模块

所需时间：3周

* 1. 参考：<http://www.cnblogs.com/zhengrui0452/p/6653964.html>

1. **编码规范**
   1. 根据结对阅读源码的经验，大家已经意识到编码规范的重要性。
   2. 讨论制定团队的编码规范，满足代码风格规范和代码设计规范（参考书第4章4.1-4.3内容）<http://www.cnblogs.com/xinz/archive/2011/11/20/2255971.html>

1.代码风格

1.1缩进

代码缩进要求：4个空白符；

所有块状结构都要求缩进，switch语句缩进要合理；

坚持一致原则，维护同学的代码时，应该保持原有的编码风格；

1.2空格

注释符两边的空白符必不可少；

任何保留字与紧随其后的左括号之间要有一个空格；

任何保留字与其前面的右大括号之间要有一个空格；

在任何二元或三元运算符的两侧必须有空格；

一元运算符与操作对象之间不允许有空格；

在, ： ； 及右括号 ) 后必须空格

数组初始化中，大括号内的空格是可选的；

行尾不得有多余的空格；

代码的水平对齐需要有；

1.3空行

类内连续成员之间，必须有1个空行；

类内的第一个成员前或最后一个成员后的空行是可选的；

没有必要添加多个连续的空行；

在函数体内，语句的逻辑分组间可使用空行；

1.4换行

项目中推荐100为字符个数的列限制；

自动换行的基本准则是：更倾向于在更高的语法级别处断开；

如果在非赋值运算符（±\*/）处断开，那么在该符号前断开；

如果在赋值运算符（==）处断开，通常的做法是在该符号后断开

自动换行时函数名与左括号留在同一行；

自动换行时逗号与前面的内容留在同一行；

自动换行时第一行后的每一行至少比第一行多缩进4个空格；

左大括号前换行，在左大括号后换行；

右大括号前换行，如果右大括号是一个语句、函数体或类的终止；否则不换行。例如，如果右大括号后面是else或逗号，则不换行。

函数的返回值不可以用括号包住，会降低代码可读性

2.代码规范

2.1命名采用普遍的驼峰命名法

2.2单行注释使用// 多行注释使用/\*

2.3在算法复杂的函数或者行列可以进行单独解释

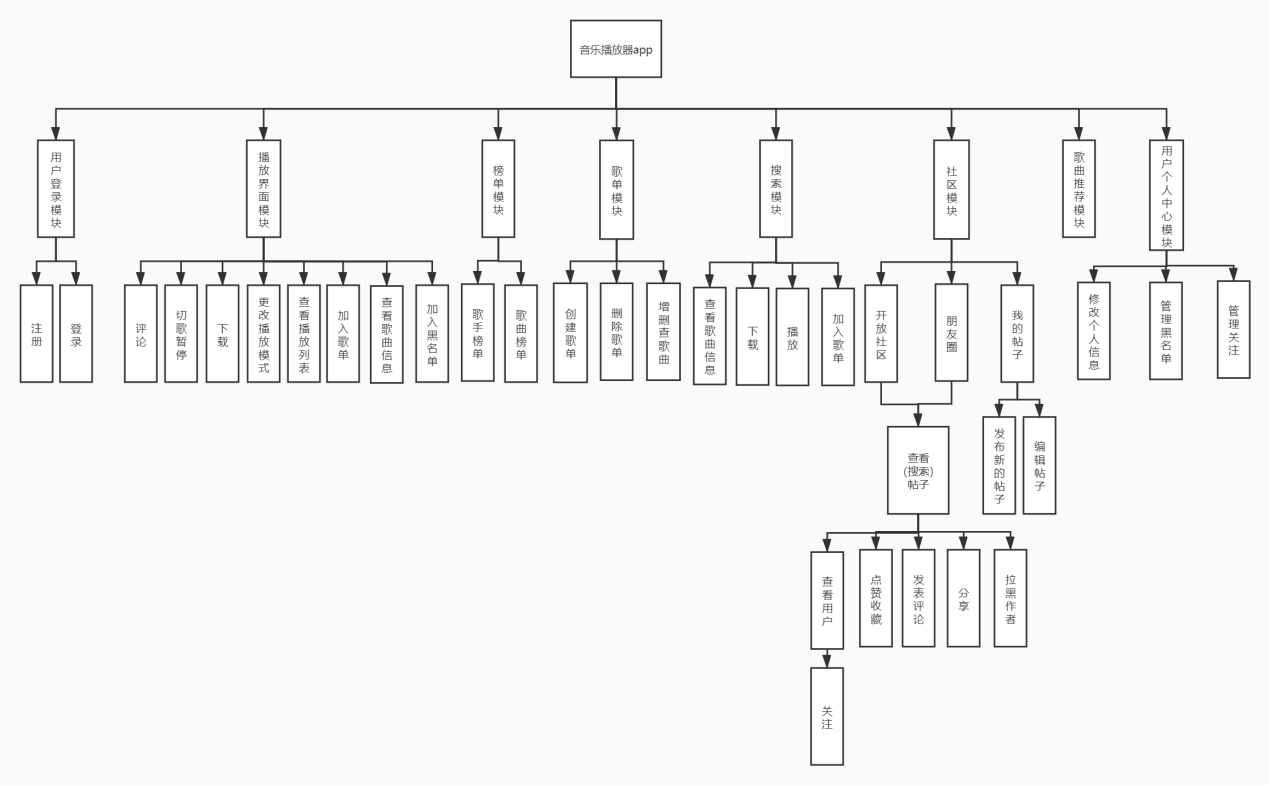
2.4注释一般不包含语言本身的语法、语言内置的API的说明、第三方类库（如Spring等）某个函数的用法的说明；不懂的去看相应的开发文档

2.5变量声明需要简介易懂，减少使用a，b，c等诸如此类的变量，变量声明在需要的时候声明，不需要把所有变量都放在函数或者代码段的开头，减少相应的维护成本，并且也同时易于优化

2.6每行尽量就使用一个语句，除非对变量进行赋值等简易操作，禁止连续赋值，尽量避免深层嵌套

2.7函数的功能尽量做得简单，每一个函数言简意赅，函数的数量控制在5-8个之间

1. **系统设计**
   1. 给出本团队项目的**体系结构图**

****

* 1. 完成团队项目的**数据库设计**，并在随笔中提供相应ER图（如果必要）

1. 数据表：

表1. 用户表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***列名*** | ***数据类型*** | ***可否为空*** | ***说明*** | ***是否为主键*** | ***是否是外键*** |
| user\_id | Char（10） | not null | 用户id | 是 |  |
| user\_name | Varchar（20） | not null | 用户昵称 |  |  |
| user\_sex | Char（2） | not null | 用户性别 |  |  |
| user\_passwd | Vchar（10） | not null | 用户密码 |  |  |
| user\_face | Blob | not null | 用户头像 |  |  |
| user\_address | Vchar（40） | not null | 用户住址 |  |  |
| user\_tel | Char（11） | Not null | 用户电话 |  |  |
| user\_email | Vchar（15） | Not null | 用户邮箱 |  |  |
| user\_birthday | Date | Not null | 用户生日 |  |  |
| isVip  user\_personalProfile | Boolean  Text | Not null  Not null | 是否为VIP  用户个人简介 |  |  |

表2. 音乐表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***列名*** | ***数据类型*** | ***可否为空*** | ***说明*** | ***是否为主键*** | ***是否是外键*** |
| music\_id | Char（10） | not null | 音乐ID | 是 |  |
| music\_name | Varchar（20） | not null | 音乐名称 |  |  |
| music\_resources | Varchar（50） | not null | 音乐资源 |  |  |
| album\_id | Char（10） | not null | 作者ID |  |  |
| isVIP\_get | Boolean | not null | 是否是VIP专享 |  |  |
| music\_like | Int | not null | 喜欢用户数 |  |  |
| music\_type | Vchar（15） | Not null | 音乐类型（分区） |  | 是 |
| music\_hearNum | Int | Not null | 音乐收听量 |  |  |
| music\_album | Char（10） | Not null | 专辑ID |  |  |

表3. 专辑表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***列名*** | ***数据类型*** | ***可否为空*** | ***说明*** | ***是否为主键*** | ***是否是外键*** |
| album\_id | Char(10) | not null | 专辑id | 是 |  |
| album\_name | Varchar(20) | not null | 专辑名称 |  |  |
| author\_id | Char(10) | not null | 作者id |  | 是 |
| music\_face | Blob | not null | 专辑封面 |  |  |
| album\_time | Date | not null | 音乐专辑发布时间 |  |  |
| album\_introduce | Text | Not null | 专辑介绍 |  |  |
| isVIP\_have | Boolean | Not null | 是否VIP专享 |  |  |
| album\_likes | Int | Not null | 喜欢专辑人数 |  |  |

表4. 歌曲评论表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***列名*** | ***数据类型*** | ***可否为空*** | ***说明*** | ***是否为主键*** | ***是否是外键*** |
| comment\_id | Char(10) | not null | 歌曲评论ID | 是 |  |
| comment\_userId | Char(10) | not null | 歌曲评论者ID |  | 是 |
| comment\_musicId | Char(10) | not null | 歌曲评论音乐ID |  | 是 |
| comment\_time | Date | not null | 歌曲评论时间 |  |  |
| comment\_identification | Int | Not null | 歌曲评论点赞数 |  |  |
| comment | Text |  | 歌曲评论内容 |  |  |

表5. 歌单表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***列名*** | ***数据类型*** | ***可否为空*** | ***说明*** | ***是否为主键*** | ***是否是外键*** |
| songSheet\_id | Char(10) | not null | 歌单id | 是 |  |
| songList | Text | not null | 歌单曲目内容 |  |  |
| owner\_id | Char(10) | not null | 拥有着ID |  | 是 |
| create\_time | Date | Not null | 创建歌单时间 |  |  |

表6. 分区表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***列名*** | ***数据类型*** | ***可否为空*** | ***说明*** | ***是否为主键*** | ***是否是外键*** |
| section\_id | Char(10) | not null | 分区id | 是 |  |
| section\_name | Varchar(20) | not null | 分区名字 |  |  |
| section\_songList | Text | Not null | 分区下的音乐列表 |  |  |

表7. 歌手表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***列名*** | ***数据类型*** | ***可否为空*** | ***说明*** | ***是否为主键*** | ***是否是外键*** |
| singer\_id | Char(10) | not null | 歌手id | 是 |  |
| singer\_name | Varchar(20) | not null | 歌手姓名 |  |  |
| singer\_indruce | Text | not null | 歌手介绍 |  |  |
| singer\_songs | Text | Not null | 歌手曲目 |  |  |

表8. 用户关注表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***列名*** | ***数据类型*** | ***可否为空*** | ***说明*** | ***是否为主键*** | ***是否是外键*** |
| uid | Char(10) | not null | 关注人id |  | 是 |
| otherId | Char(10) | Not null | 被关注人id |  | 是 |

表9. 帖子表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***列名*** | ***数据类型*** | ***可否为空*** | ***说明*** | ***是否为主键*** | ***是否是外键*** |
| post\_id | Char(10) | not null | 帖子id | 是 |  |
| post\_name | Varchar(20) | not null | 帖子标题 |  |  |
| post\_creatTime | Date | not null | 帖子发布时间 |  |  |
| post\_authorId | Char(10) | Not null | 帖子作者id |  | 是 |
| post | Text |  | 帖子内容 |  |  |

表10. 帖子评论表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***列名*** | ***数据类型*** | ***可否为空*** | ***说明*** | ***是否为主键*** | ***是否是外键*** |
| post\_comment\_date | Date | not null | 帖子评论时间 |  |  |
| post\_comment\_id | Char(10) | not null | 帖子评论id | 是 |  |
| post\_id | Char(10) | not null | 评论的帖子id |  | 是 |
| post\_comment | Text | Not null | 评论内容 |  |  |
| post\_comment\_userId | Char(10) |  | 帖子评论者id |  | 是 |

表11. 帖子点赞表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***列名*** | ***数据类型*** | ***可否为空*** | ***说明*** | ***是否为主键*** | ***是否是外键*** |
| post\_id | Char(10) | not null | 帖子id |  | 是 |
| post\_like\_id | Char(10) | Not null | 点赞者id |  | 是 |

表12. 歌曲点赞表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***列名*** | ***数据类型*** | ***可否为空*** | ***说明*** | ***是否为主键*** | ***是否是外键*** |
| music\_id | Char(10) | not null | 歌曲id |  | 是 |
| music\_like\_userId | Char(10) | Not null | 点赞者id |  | 是 |

表13. 帖子拉黑表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***列名*** | ***数据类型*** | ***可否为空*** | ***说明*** | ***是否为主键*** | ***是否是外键*** |
| post\_id | Char(10) | not null | 帖子id |  | 是 |
| post\_unlikeId | Char(10) | Not null | 帖子拉黑者id |  | 是 |

表14. 拉黑歌曲表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***列名*** | ***数据类型*** | ***可否为空*** | ***说明*** | ***是否为主键*** | ***是否是外键*** |
| music\_id | Char(10) | not null | 音乐id |  | 是 |
| music\_unlikeId | Char(10) | Not null | 音乐拉黑者id |  | 是 |

表15. 关注音乐表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***列名*** | ***数据类型*** | ***可否为空*** | ***说明*** | ***是否为主键*** | ***是否是外键*** |
| music\_id | Char(10) | not null | 音乐id |  | 是 |
| music\_followId | Char(10) | Not null | 关注者id |  | 是 |

表16. 关注帖子表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***列名*** | ***数据类型*** | ***可否为空*** | ***说明*** | ***是否为主键*** | ***是否是外键*** |
| post\_id | Char(10) | not null | 帖子id |  | 是 |
| post\_followId | Char(10) | Not null | 帖子关注者id |  | 是 |

表17. 关注歌手表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***列名*** | ***数据类型*** | ***可否为空*** | ***说明*** | ***是否为主键*** | ***是否是外键*** |
| singer\_id | Char(10) | not null | 歌手id |  | 是 |
| follow\_singer\_userId | Char(10) | Not null | 歌手关注者id |  | 是 |

表18. 关注歌单表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***列名*** | ***数据类型*** | ***可否为空*** | ***说明*** | ***是否为主键*** | ***是否是外键*** |
| singSheet\_id | Char(10) | not null | 歌单id |  | 是 |
| follow\_singSheet\_userId | Char(10) | Not null | 歌单关注者id |  | 是 |

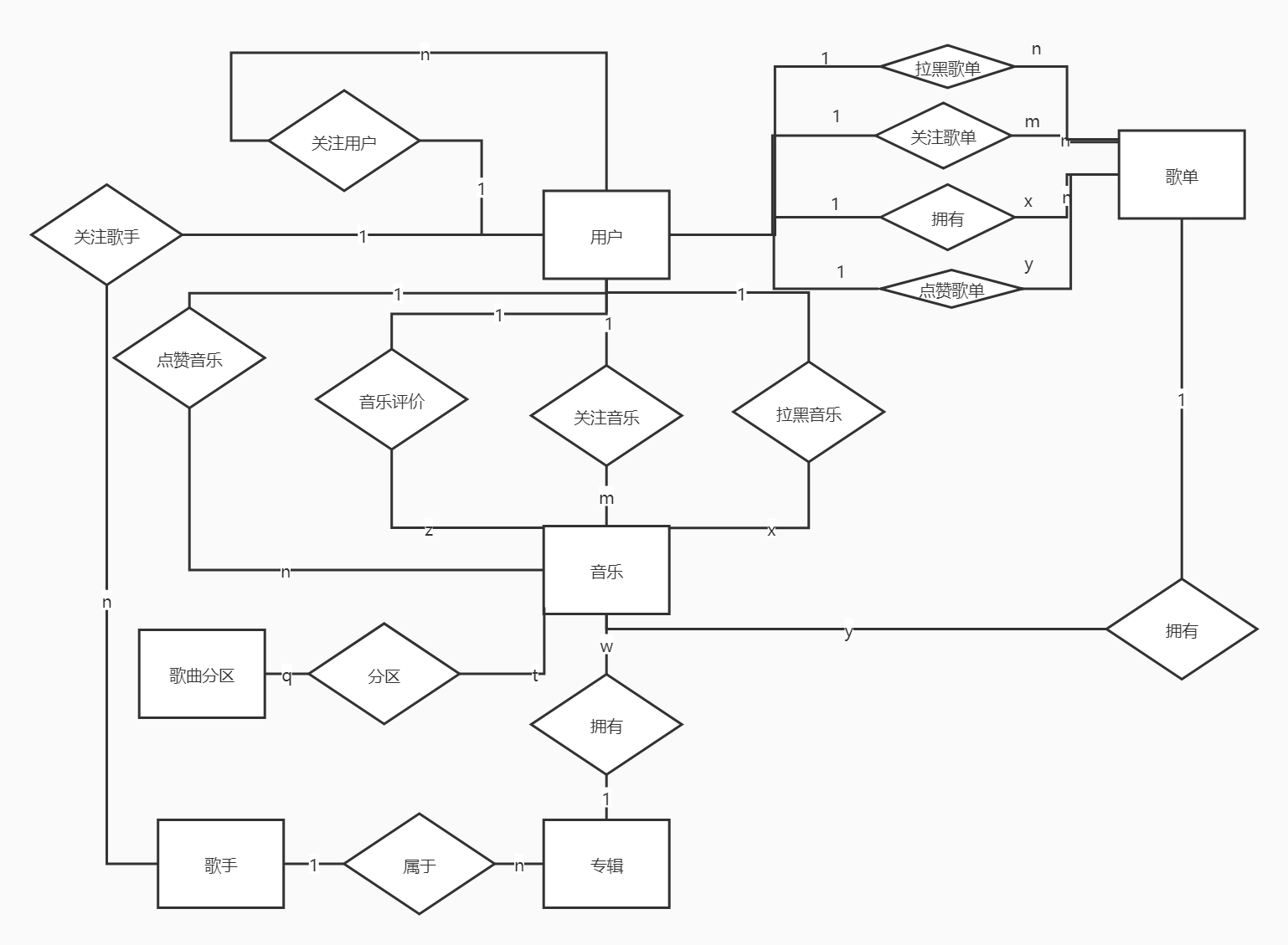
表19. 拉黑歌单表

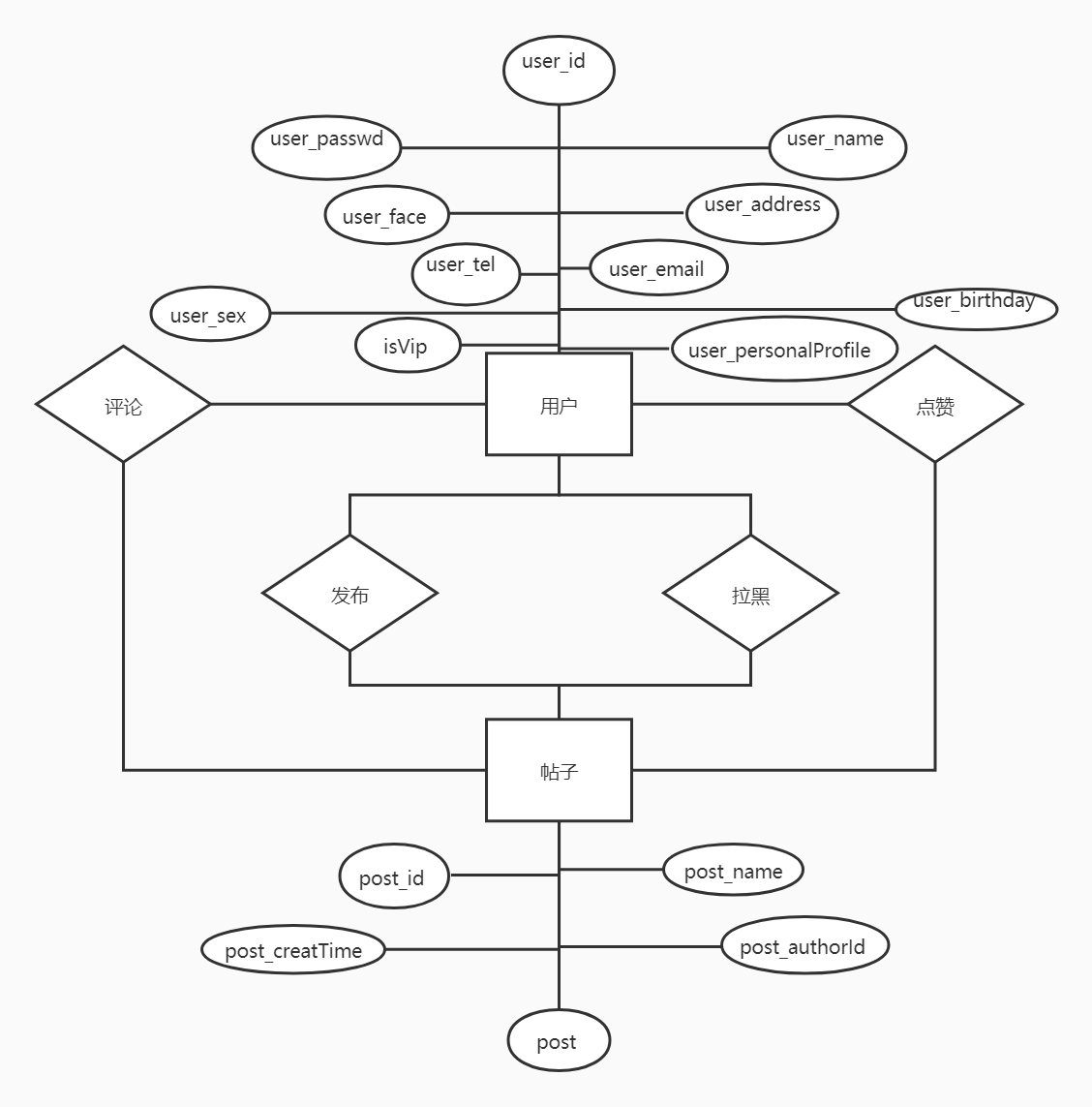
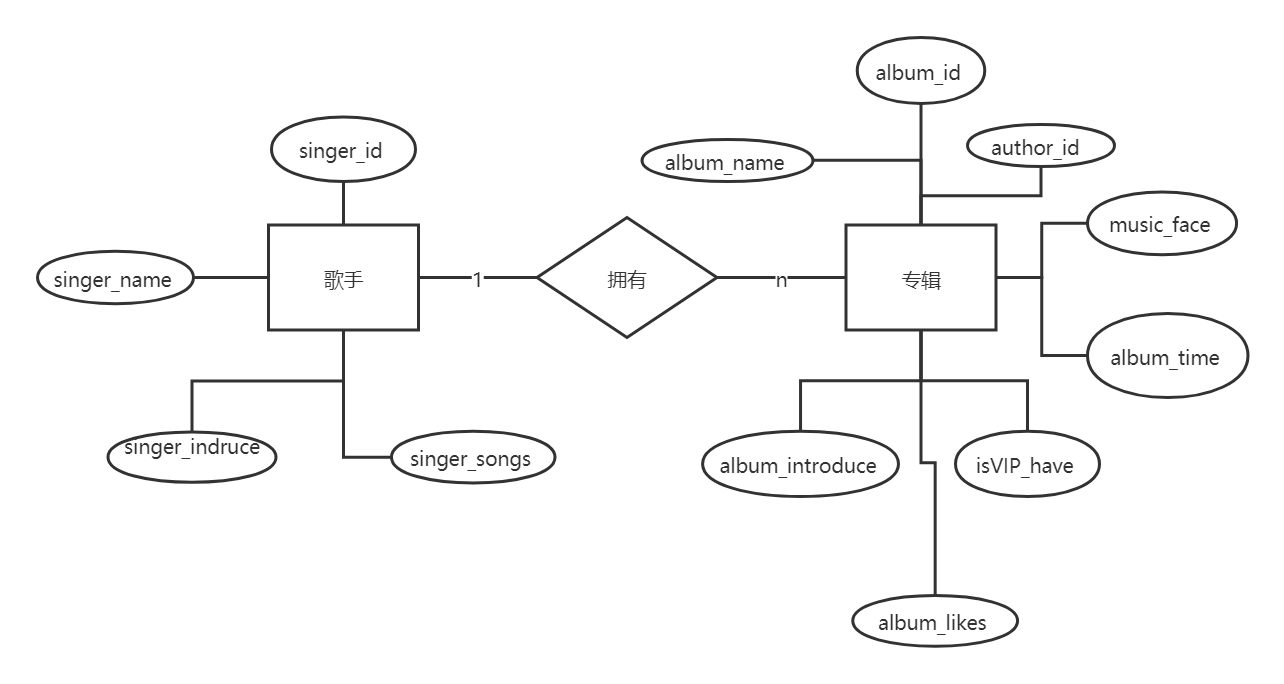
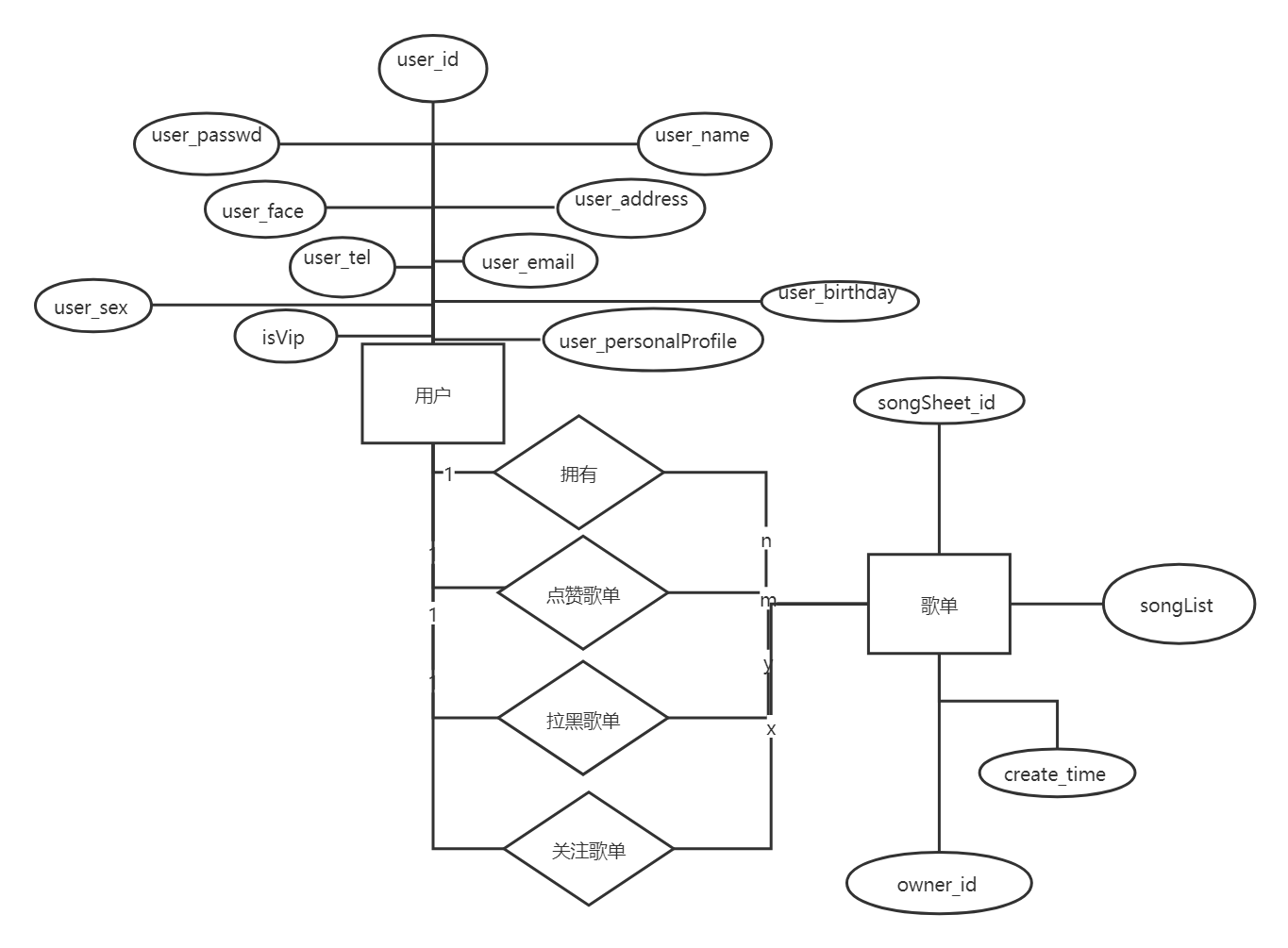
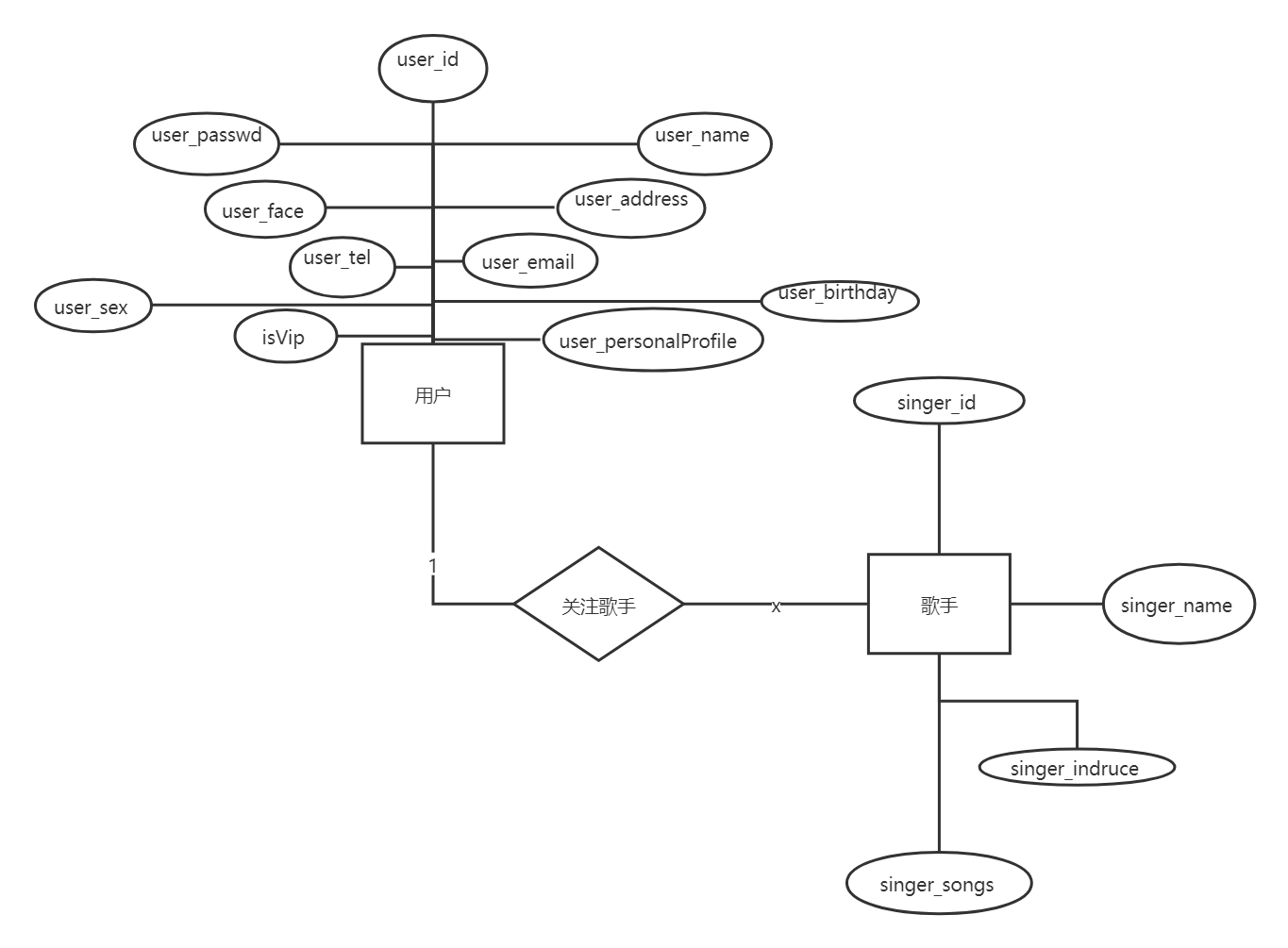
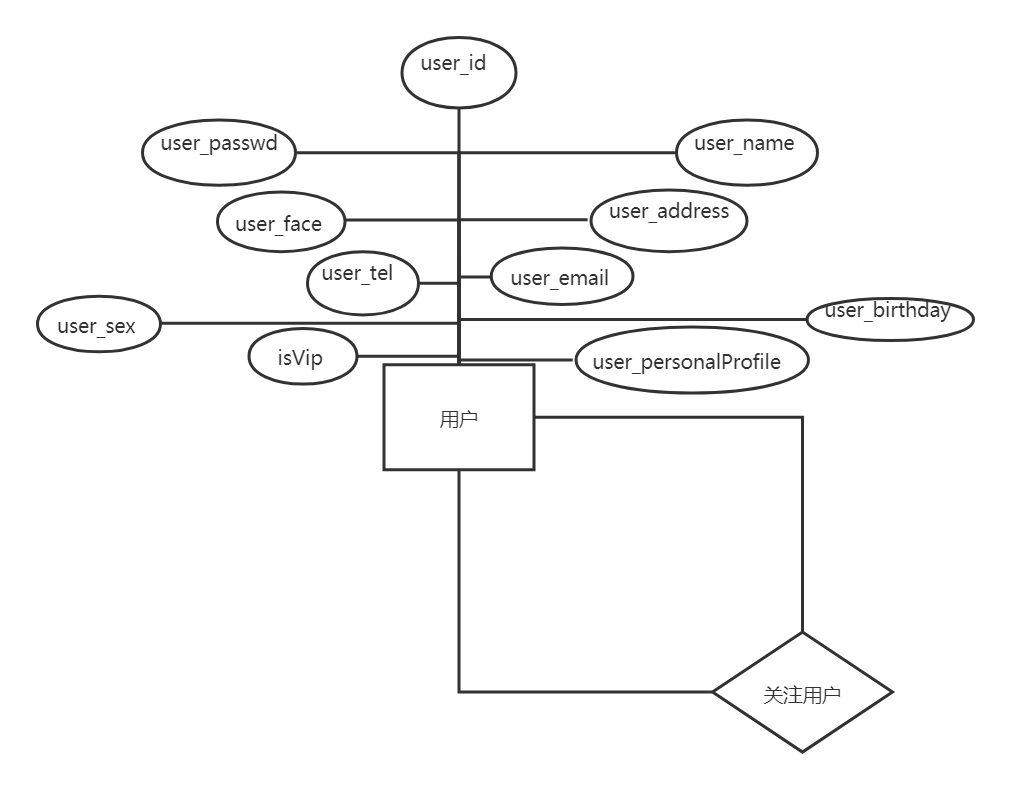
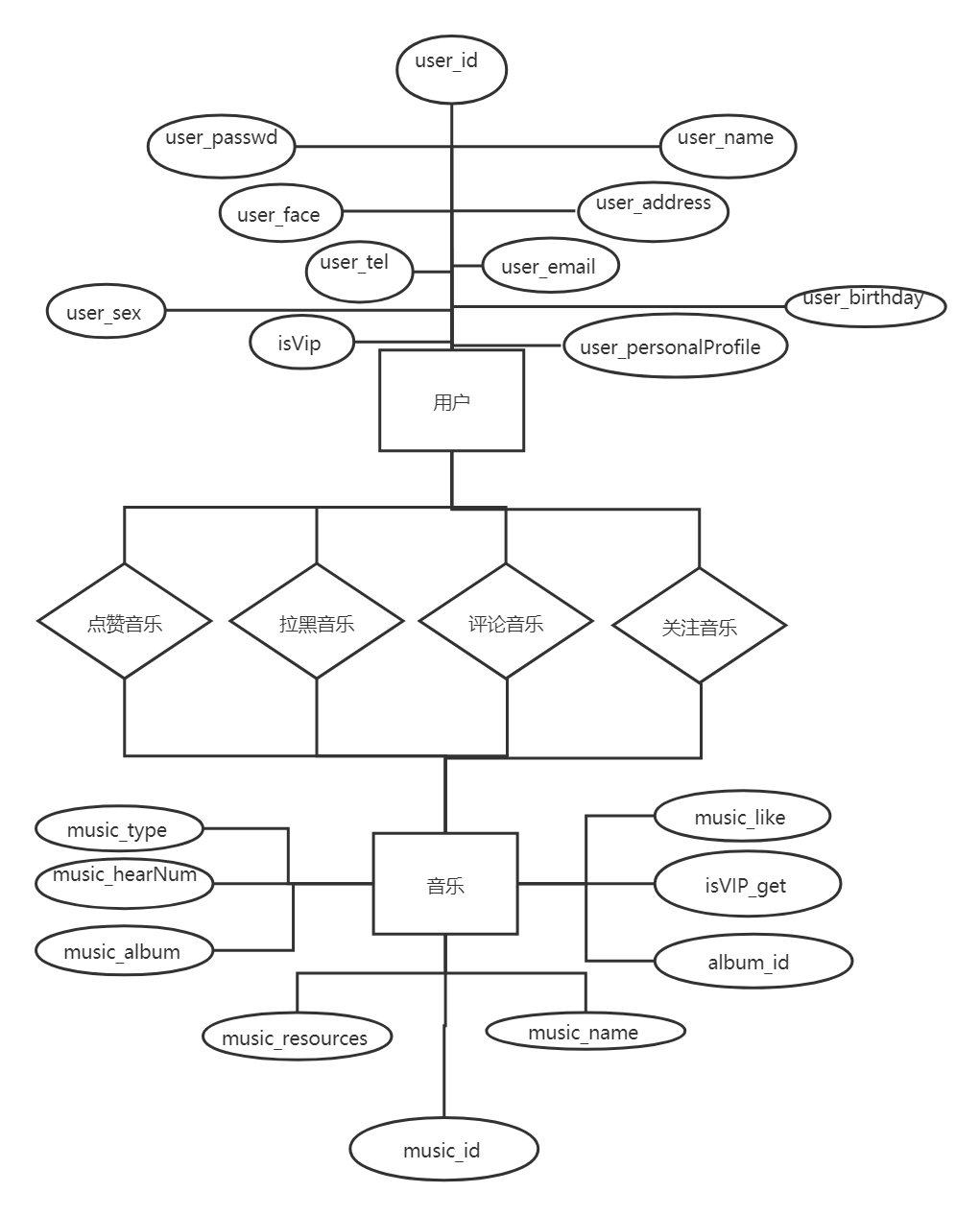
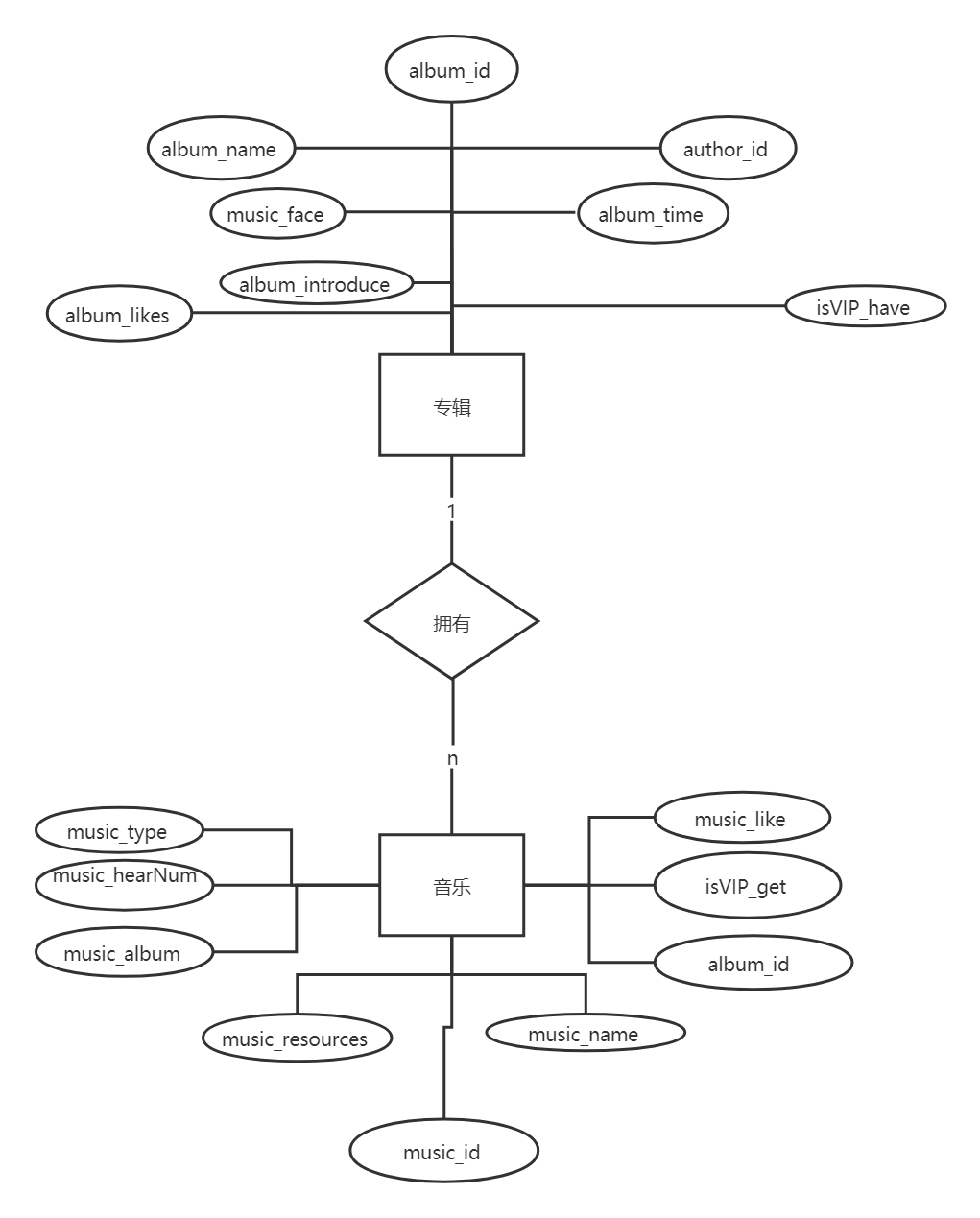
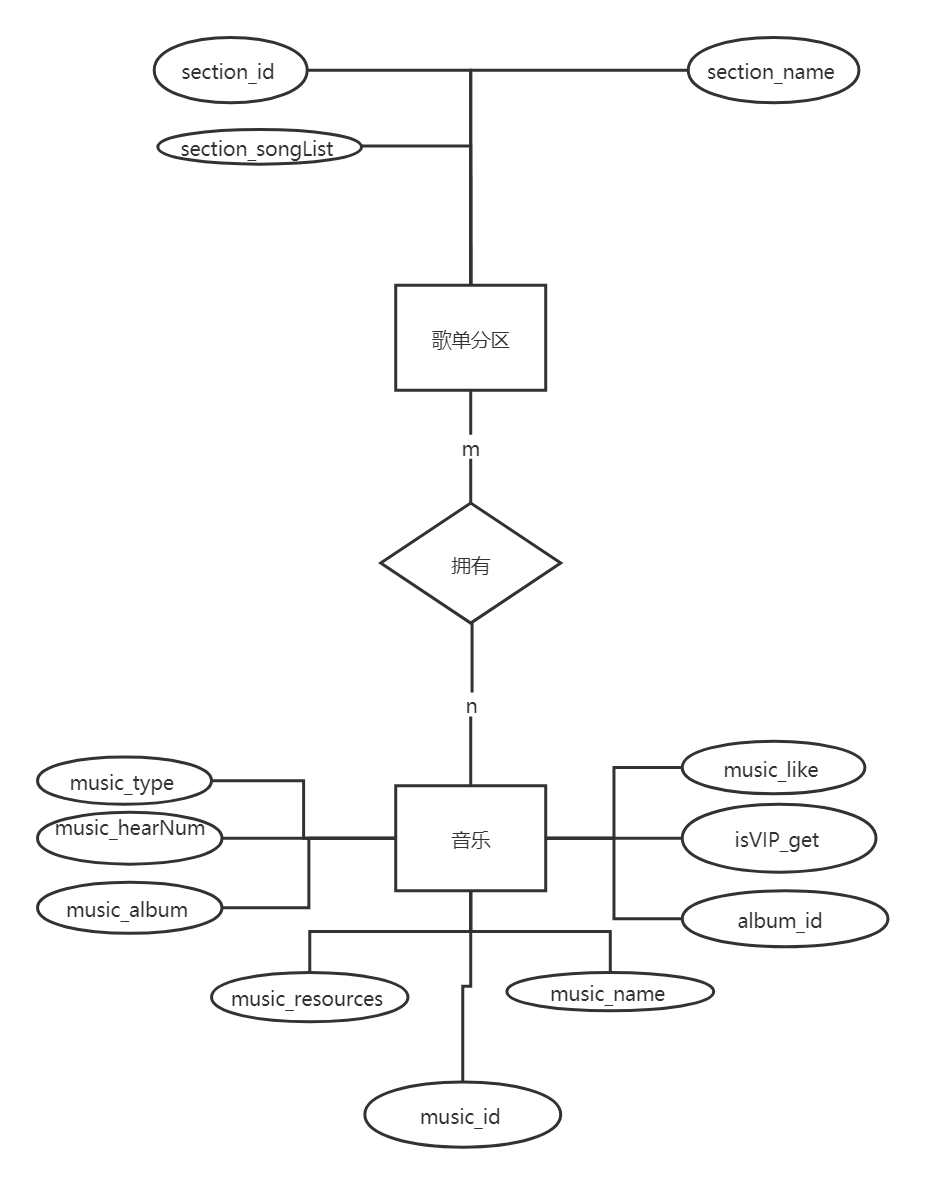
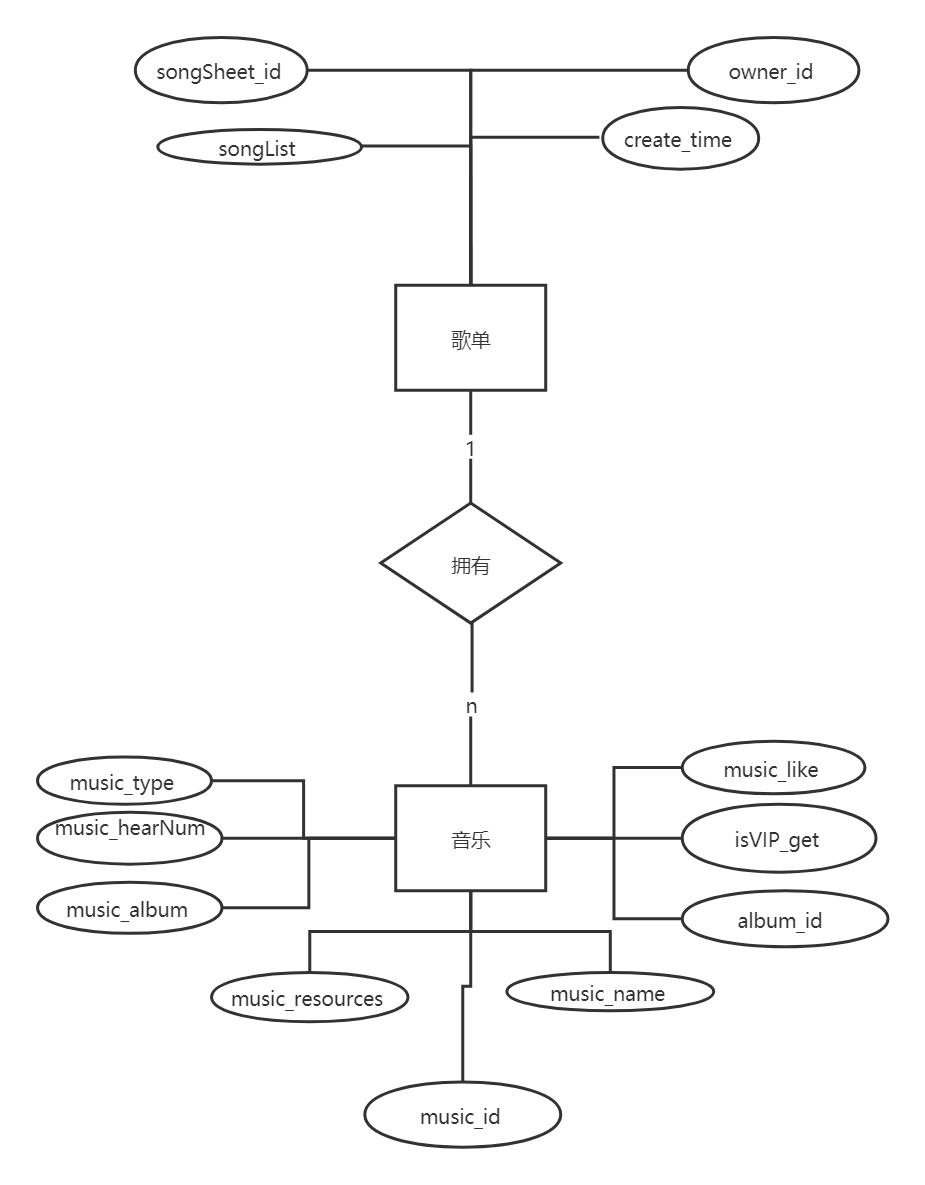
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***列名*** | ***数据类型*** | ***可否为空*** | ***说明*** | ***是否为主键*** | ***是否是外键*** |
| singSheet \_id | Char(10) | not null | 歌单id |  | 是 |
| unlike\_singSheet\_userId | Char(10) | Not null | 歌单拉黑者id |  | 是 |

表20. 点赞歌单表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***列名*** | ***数据类型*** | ***可否为空*** | ***说明*** | ***是否为主键*** | ***是否是外键*** |
| singSheet\_id | Char(10) | not null | 歌单id |  | 是 |
| like\_singSheet\_userId | Char(10) | Not null | 歌单点赞者id |  | 是 |

部分重要E-R图：





* 1. 案例参考：https://www.cnblogs.com/LDLYMteam/p/8696231.html

1. 报告按时提交课堂派。