

主管  
领导  
审核  
签字

软件工程 试 题

题号	一	二	三	四	五	六	总分
得分							
阅卷人							

注：本试卷满分 100 分，折合为 40 分计入总成绩。

年轻人喜欢随时随地享受音乐，但每个人对音乐的兴趣是不同的。拟开发一个基于互联网的音乐推荐器，根据用户听歌的历史记录，为个人推荐符合其兴趣的音乐。该软件的意义在于：一方面为用户提供个性化的音乐服务，另一方面帮助唱片公司推广新音乐作品，找到潜在听众。

该音乐推荐器为三类用户提供服务：个人听众、电台 DJ、唱片公司，这三类用户的重要度从高到低排序。音乐推荐器需维护一个庞大的音乐库，包含很多歌曲，但该库仅存储歌曲的在线播放 URL，并非实际的可播放 mp3 文件。每首歌曲都有一组标签，每个标签用于表征该歌曲在某一方面的属性(如古典、乡村、中国好声音、伤感、交响乐等)，用于后续推荐。不过，从 0 开始构建音乐库的难度较大，最好是将该软件与百度音乐的音乐库连接起来，从中读取音乐数据并同步到本软件中。

个人听众上传自己喜欢的音乐清单，系统根据该清单以及该听众的历史听歌记录，分析发现该听众的潜在兴趣(亦由一组标签构成)，进而使用特定的推荐算法，从音乐库里选择出最符合其兴趣的一组歌曲推荐给该听众。听众在线欣赏这些歌曲的过程中，可随时标注对软件推荐的某首歌曲是“喜欢”还是“不喜欢”，系统据此动态调整该听众的兴趣，用于更新对该听众的后续推荐。

电台 DJ 利用该软件进行电台节目中所播歌曲的选择。他向软件提供自己感兴趣的若干主题(由一组标签构成)，系统根据这些主题与歌曲标签的匹配、过去一段时间内特定听众群体的听歌记录，推荐这些听众最喜欢的若干歌曲给 DJ。DJ 从中手工选择一部分歌曲，软件将所选歌曲的在线 URL 生成特定类型的文件(如 txt、xls、html 等)并 email 给他。

唱片公司的发行系统与该音乐推荐器相连接，可将该公司拟发布的新歌分享到音乐推荐器。该软件将新歌加入到音乐库中，根据全体听众/电台 DJ 在过去一段时间内所欣赏与选择的歌曲，估算出该首新歌的受欢迎程度，向唱片公司推荐出对该首新歌最可能感兴趣的一组听众，同时以实时推送的形式将新歌推荐给这些潜在听众。

该音乐推荐器计划采用两种形式：移动端 App、Web 网站。从 NFR 角度看：(1) 由于该软件是典型的互联网服务，音乐库数据的更新和同步操作将非常频繁；(2) 听众/DJ/唱片公司使用该软件时希望具备可定制性(个性化)；(3) 在某些时段(如午后和午夜)，用户的访问量非常大，软件在高并发情况下应保持平稳性能(较高的可用性和访问速度等)。

一 (15 分)需求获取与敏捷开发

在第一列列出该软件的全体参与者，分析各自的用例清单，并评估各自的优先级(5、4、3、2、1)。用例的名称请采用简洁明了的中文短语，每行填写一个。下表各行未必全部填满。

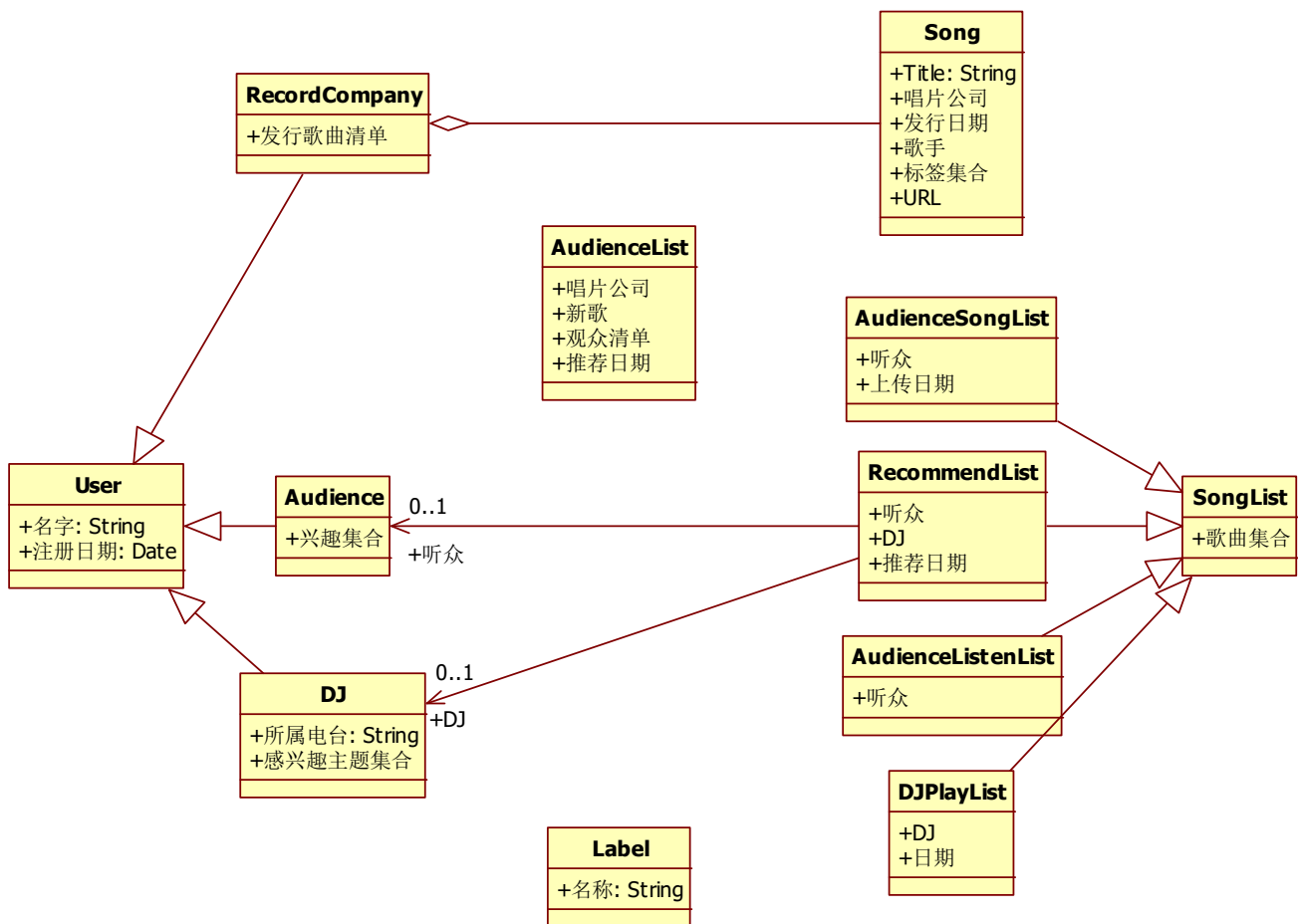
参与者名称	用例	优先级估算

## 二(25分)OO 分析与设计

该软件采用 OO 方式进行分析与设计。已从需求描述中识别出了一组实体类，其简要描述和关键属性如下表所示。

实体类	说明	属性
User	以下三个类的父类	名字、注册日期
Audience	使用该软件的个人听众	个人兴趣集合
DJ	使用该软件的电台 DJ	所属电台、感兴趣的 主题集合
RecordCompany	使用该软件的唱片公司	发行歌曲的清单
Label	对歌曲分类的标签，用于标注各首歌曲 的属性、听众的兴趣、DJ 感兴趣的 主题	名称
Song	该软件所管理和使用的歌曲	Title、唱片公司、发 行日期、歌手、标 签集合、URL
SongList	以下四个类的父类	歌曲集合
AudienceSongList	听众上传的音乐清单，软件仅读取并 用作推荐，不对其 进行更新	听众、上传日期
RecommendList	听众或电台 DJ 每次使用推荐功能之 后所获得的推荐音 乐清单	用户、推荐日期
AudienceListenList	听众在该软件中在线听歌所形成的音 乐清单，由系统自 动更新维护，每个 听众分别有一个这 样的清单	听众
DJPlayList	电台 DJ 在该软件中所选择的音乐清 单，DJ 每次使用推 荐功能并作出选择 之后均形成这样一 个清单	电台 DJ、日期
AudienceList	唱片公司发布的某首新歌从该系统 中所获得的推荐听 众清单	唱片公司、新歌、0 到多个听众所构 成的清单、推荐 日期

- (1) (20 分)下图给出了这些实体类之间的关系描述，但并不完整。请根据上表所做出的限定，直接在类图上补充缺失的类间关系(继承、聚合、组合、关联)，并补全关系上的角色名和多重性信息。不能删除图中已有信息，只能增加或修改，无需补充或修改类的属性和操作。



- (2) (5 分)在需求陈述中提到：听众在线欣赏这些推荐音乐的过程中，可以随时标注对某首歌曲是“喜欢”还是“不喜欢”。该功能尚无法被上页表格所给出的实体类所支持。为了实现该功能，在已有实体类的前提下，需要引入哪个(哪些)新的实体类？给出新实体类的名称、属性集合及各属性的数据类型，以及它(们)与其他已有各实体类之间的关系(继承、关联、组合、聚合)。

姓名

学号

院系

密

封

线

### 三(20 分)软件配置管理与 Git

软件开发团队在某段时间内对源代码进行基于 Git 的软件配置管理操作如下表所示。请据此绘制出该项目的“提交”(commit)和“分支”(branch)的变迁关系图, 节点集合为 Git 产生的全部 commit 对象, 有向边为各 commit 对象之间的依存关系, 并在图上标注出每个分支(包含 HEAD)最终所指向的 commit 对象。

请在右侧空白区绘图, 采用自下而上的次序进行绘制, 处于下方的节点的提交/更新时间早, 越往上的节点的提交/更新时间越晚。

```
//create files
git init
git add *.*
git commit -m 'initial commit'
git checkout -b 'Fix1'
//modify files and git add *
git commit -m 'the first fix'
//modify files and git add *
git commit -m 'the second fix'
git checkout master
//modify files and git add *
git commit -m 'initial commit 2'
git checkout 'Fix1'
//modify files and git add *
git commit -m 'the third fix'
git branch 'Fix2'
git checkout 'Fix2'
//modify files and git add *
git commit -m 'the fourth fix'
git checkout 'Fix1'
//modify files and git add *
git commit -m 'the fifth fix'
git merge 'Fix2'
git checkout 'master'
git checkout -b 'Fix3'
//modify files and git add *
git commit -m 'the sixth fix'
git merge 'Fix1'
git merge 'master'
```

---

#### 四 (20 分)面向 NFR 的架构设计

(1) (9 分)若采用移动端 App 作为听众、DJ 和唱片公司的客户端，那么该软件所维护的庞大音乐库需如何部署，有以下三种策略：

策略 1：音乐库完全部署于 App 端，服务端为 0；

策略 2：音乐库完全部署于服务器端，App 端为 0；

策略 3：完整的音乐库被部署于服务器端，App 端仅部署该用户感兴趣的音乐子库，并随着用户使用该软件而动态更新。

请分析这三种部署策略在哪些非功能指标上存在折中、它们在每一项指标上的优势和劣势分别是什么？

姓名

学号

院系

密

封

线

(2) (5 分)需求中提到:“该软件与百度音乐的音乐库连接起来,从中读取音乐数据并同步到本软件中”。现为该需求设计相应的接口,其功能是支持音乐数据的同步。考虑以下两项冲突的 NFR: (a) 百度音乐库中的数据在频繁更新,故该软件内的音乐信息应尽可能与百度音乐库保持实时的一致性; (b) 同步时应尽量降低对该软件运行性能的影响,故同步的频度和数据量应尽可能低。请给出该接口的设计思路,并阐述该设计如何能够实现上述 NFR。

(3) (6 分)需求中提到:“在某些时段(如午后和午夜),用户的访问量非常大,软件在高并发情况下应保持平稳性能(较高的可用性和访问速度等)”。请给出至少 2 项架构设计决策来支持上述 NFR,并阐述相应的依据。请不要简单给出通用的架构设计决策,务必仔细考虑该软件的特征。

序号	设计决策	为何可支持该项 NFR
1		
2		
3		

---

## 五 (10 分)软件测试

“小孩子才分对错，成年人只看利弊” ——电影《后会无期》

该软件的难点是各种推荐算法。以算法“针对听众在该软件内的听歌记录，为该听众推荐音乐”为例，其输入是“某听众的历史听歌记录”，其输出是“一组推荐音乐”。与传统的黑盒测试不同，该推荐结果无法被客观的判定为“正确”还是“错误”，而需要由听众来判定该推荐结果符合自己的兴趣期望的程度。因此，仅依靠算法输出的“一组推荐音乐”无法进行测试。

若开发团队已经实现了该算法，正在对其进行黑盒测试，设计了一组测试用例并加以实际执行。简要阐述：

- (1) (4 分)除了算法直接输出的“一组推荐的音乐”，还需要从测试执行过程中采集什么数据，才能有效判定该算法设计的优劣？
- (2) (3 分)根据所采集到的数据对算法进行优劣评判的标准是什么？
- (2) (3 分)针对该推荐算法，如何评判所设计的一组测试用例是充分的？



姓名

学号

院系

密

封

线

## 六 (10 分)观点论述

以下题目 4 选 2 作答，每题 5 分。答题时请标注清楚你所选的题号。若回答超过 2 题，按前 2 个答题计分。

- (1) “敏捷开发方法要求团队成员为‘全才’，技术水平应足够高，否则团队无法敏捷起来”。该观点是否正确？请给出理由。
- (2) 图灵奖得主 Fredrick Brooks 在其《人月神话》一书中提出一个观点：“Adding manpower to a late software project makes it later”。在当前的主流软件开发模式下，该观点是否仍然成立？简要阐述其原因。
- (3) 在面向对象的设计原则 SOLID 中，“依赖转置原则(DIP)”与“开放封闭原则(OCP)”之间有什么内在的联系？
- (4) 在软件测试过程中，“测试用例的数目”、“测试的覆盖度”、“测试的效率”三者之间存在一定的关系。简要分析它们之间的折中性。

---