

天津大学



《软件工程》 实验说明书

在线歌曲播放器

2021 年 6 月 26 日

学院：智能与计算学部

班级：网络安全一班

姓名：谢远峰

学号：3019244283

方法：面向对象软件工程设计

摘要

本次项目按照面向对象软件工程的流程进行分析,从选题、需求分析、概念设计、详细设计、系统关键样图实现、软件测试、项目安装与部署,通过线下图书资料的查阅和线上资源收集整理,完成本文档的撰写,系统的设计开发和实现流程视频讲解。

本文档共分为八大部分,从传统软件工程设计角度覆盖本次项目的全部流程,包括项目说明需求分析,概要设计,详细设计,详细设计,系统实现,软件测试,软件安装部署,项目总结。

项目说明阐述了本次项目的基本信息,并对项目的设计目的进行概括性描述,包括项目名称、项目简介、项目的核心功能介绍。

需求分析从实际的生活场景出发,发现项目的涉众,并进行涉众群体特征分析,确定本项目的功能性需求(通过用例图进行描述),同时添加非功能性需求,并从可靠性,安全性,性能和界面多个方面进行阐述。

概要设计从系统架构入手,选择 C/S 架构作为本项目的系统架构,并基于需求部分的相关需求分析设计了系统的主要功能模块并进行深度的解释说明,绘制相关的系统用例图(及用例规约表)和主要的活动图使得项目需求更加清晰明确。

详细设计基于概要设计部分的分析结果,从编程角度出发,绘制类图与包图,主要的序列图状态图,同时进行数据设计,帮助自身理解项目的相关内容,为编程实现确立清晰的架构。

系统实现包含项目关键界面的介绍,其中包含用户注册页面,用户登录页面网站首页,歌曲的播放页面。

软件测试分析设计本项目的软件测试计划,包含单元测试,系统测试,并搭配相对应的测试截图。

软件安装部署对项目部署的相关信息阐述,包含部署内容,部署范围,部署环境(硬件环境、软件环境、网络环境)等部分。

项目总结该实验报告从传统软件功臣的角度对软件的开发进行由粗到细的梳理,对于课上习得的知识形成进一步的了解和掌握,并将所学知识结合实际软件开发进行深度融合,针对开发中遇到的实际问题进行灵活多变的处理。

目录

1 项目说明	3
1.1 项目名称	3
1.2 项目简介	3
1.3 核心功能	3
2 需求分析	3
2.1 项目背景	3
2.2 项目目标	3
2.3 涉众分析	3
2.4 功能性需求	4
2.4.1 系统用例图	4
2.4.2 系统用例描述	5
2.5 包图	6
2.6 用例规约表	6
2.7 非功能性需求	7
3 概要设计	8
3.1 页面设计	8
3.1.1 网站首页	8
3.1.2 歌曲排行榜页面	8
3.1.3 歌曲播放页面	8

3.1.4	歌曲点评页面	8
3.1.5	歌曲搜索页面	8
3.1.6	用户中心页面	9
3.2	系统架构图	9
4	详细设计	9
4.1	软件进度计划	9
5	数据库设计	10
5.1	数据表	10
5.2	数据表内容	10
5.3	功能模块设计	11
5.3.1	网站首页	11
5.3.2	歌曲排行榜页面	11
5.3.3	歌曲播放页面	11
5.3.4	歌曲点评页面	11
5.3.5	用户中心页面	12
5.3.6	网站后台	12
6	系统部署	12
6.1	部署环境	12
6.2	代码浏览	12
7	软件测试	13
7.1	单元用例测试	13
7.1.1	黑盒单元测试	13
7.1.2	白盒单元测试	14
7.2	测试截图	15
7.2.1	注册功能测试截图	15
7.2.2	登录模块测试截图	16
8	项目总结	17

1 项目说明

1.1 项目名称

歌曲在线管理系统

1.2 项目简介

歌曲在线播放网站是为用户电脑客户端开发的在线歌曲播放管理系统，系统可进行使用和管理，可进行歌曲的播放，下载，管理。

电脑网页端使得本系统具有较强的灵活性，用户只需在线打开本网页即可进行在线的歌曲收听，同时根据歌曲的类型和热度，选择适合自己的歌曲风格，较于传统的歌曲播放器界面单一化，无有效即时的交互信息反馈的痛点进行修改，使得用户能动态在网页上实现“歌曲自由”。

1.3 核心功能

1. 歌曲管理：实现歌曲的播放，歌曲歌词，歌曲作者，添加时间，歌曲时长的查看，歌曲的下载
2. 用户管理：实现用户注册，登录情况的管理
3. 歌曲管理：对于歌曲设置主题，热度，搜索量的信息，利于用户进行查询

2 需求分析

2.1 项目背景

随着互联网时代的到来，网络早已渗透进人们的日常生活中，并成为信息传播的一部分。网络资源的获取改变了人们以往的生活方式，网络成为日常工作、休闲的主要工具之一。基于网络的歌曲网站应运而生为广大的歌曲爱好者提供一个歌曲交流的平台，增加对于歌曲的了解。

用户听歌进行休闲时，希望获取到关于歌曲的相关信息，其中包括：歌曲的演唱者，歌曲的名称，歌曲的时长，歌曲所从属的专辑名称，歌曲从属的歌曲主题类型，其它用户关于该歌曲的评论。

管理者对于网站上使用的用户进行账号管理，可以对于特定用户进行添加和删除，可以添加或删除特定的歌曲，并且能够查看特定用户的播放记录。

2.2 项目目标

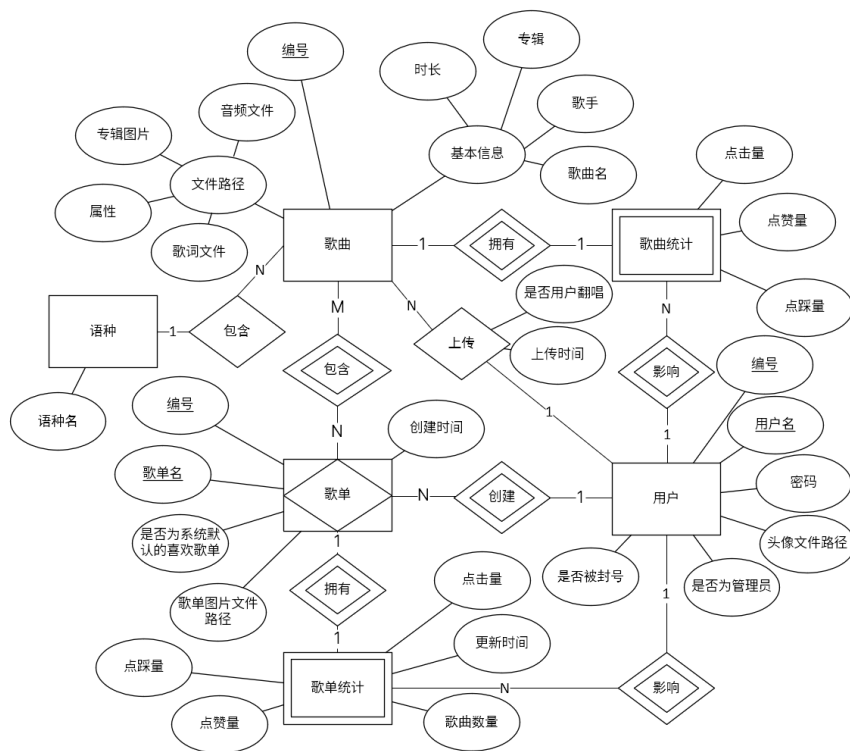
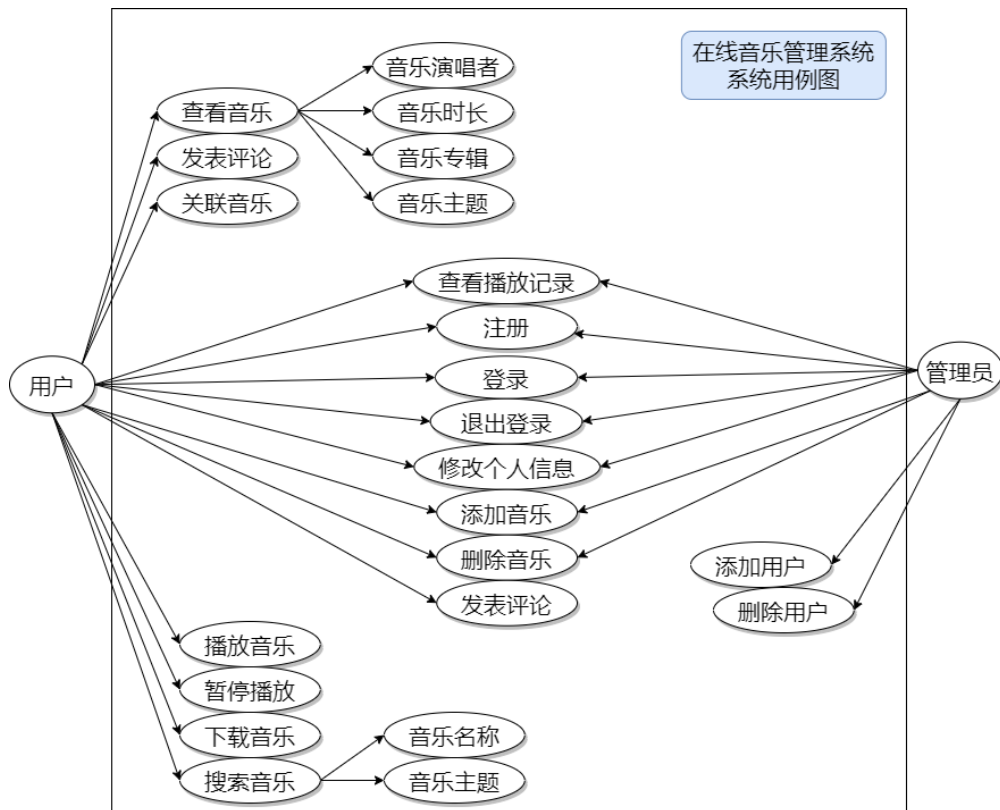
1. 针对众多歌曲进行分类管理，可随时查看自己的歌曲播放记录
2. 针对歌曲热度和搜索量，能够了解当下大众的歌曲偏好
3. 对于管理者，可查看用户的歌曲播放记录，利于进行管理

2.3 涉众分析

编号	涉众名称	涉众说明	期望
sh01	普通用户	使用歌曲网站的普通用户	1. 收听歌曲 2. 下载歌曲 3. 搜索歌曲 4. 查看歌曲信息
sh02	管理者	管理歌曲网站用户及信息的管理员	1. 添加、删除用户 2. 添加、删除歌曲 3. 展示歌曲的信息

2.4 功能性需求

2.4.1 系统用例图



2.4.2 系统用例描述

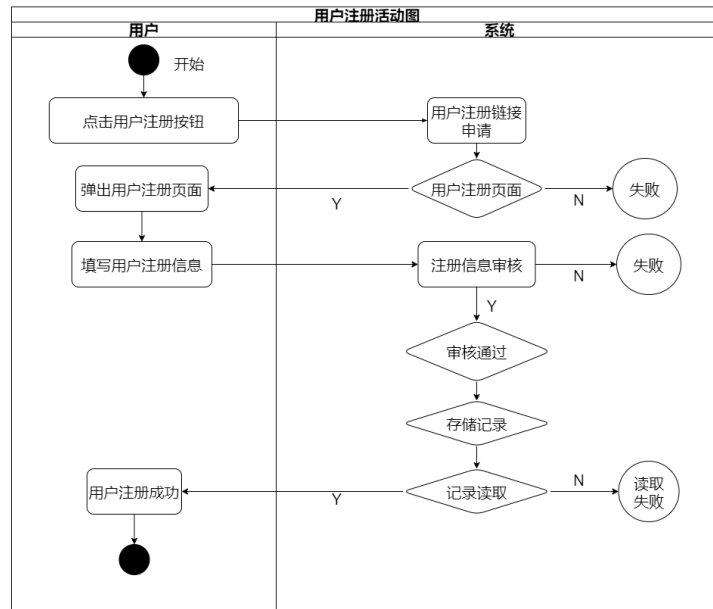


图 2.4.2.1: 登录系统用例图

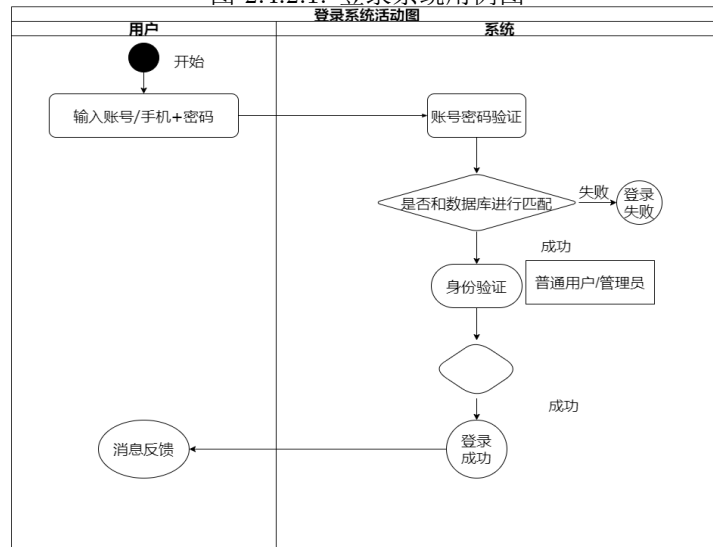


图 2.4.2.2: 登录系统用例图

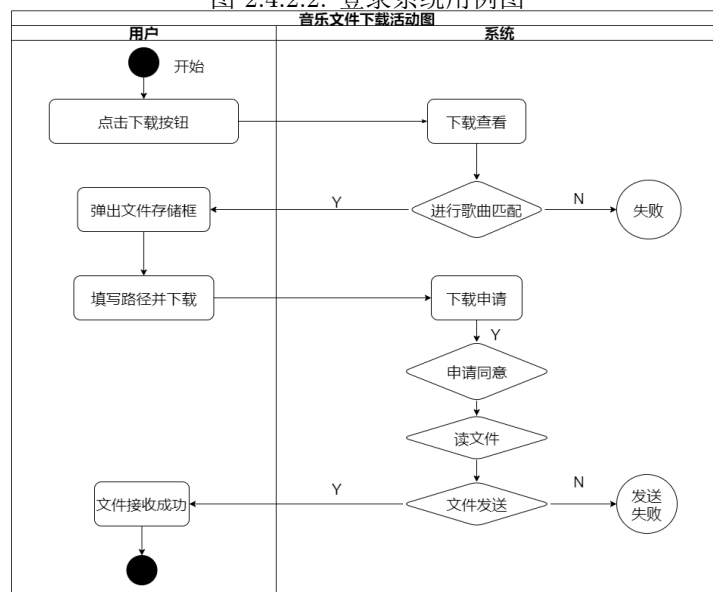


图 2.4.2.3: 音乐文件下载用例图

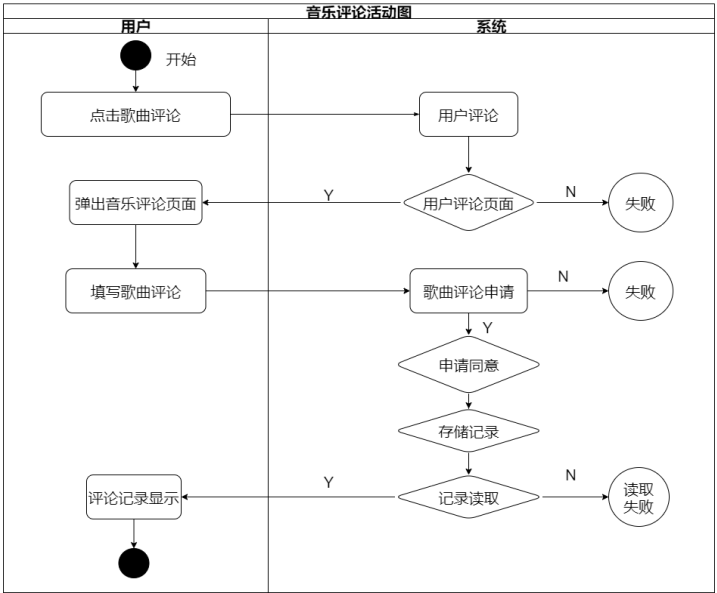


图 2.4.2.4: 音乐评论活动用例图

2.5 包图

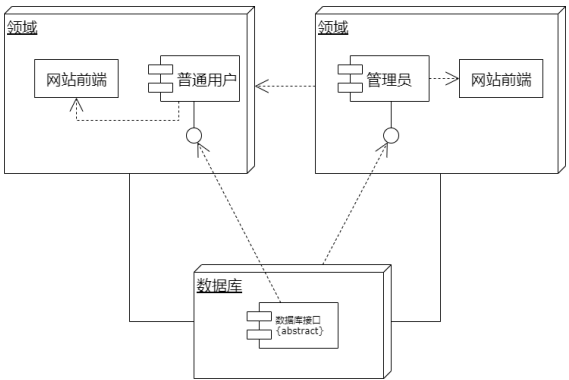


图 2.5.0.1: 音乐评论活动用例图

2.6 用例规约表

用例编号	1.1
用例名称	网站用户登录
执行者	普通用户
描述	普通用户登录网站
前置条件	进入网站下设的登录页面
基本流程	1. 用户进入指定的网址 2. 用户输入用户名/手机号，和密码 3. 进入用户管理界面
扩展流程	1. 如果用户未进入网址，则无法操作 2 2. 用户未填写或填写错误，则无法进入 3
后置条件	无

表 2.6.0.1: 普通用户登录

用例编号	1.2
用例名称	网站用户登录
执行者	管理员
描述	管理员登录网站
前置条件	进入网站后台的管理页面
基本流程	1. 用户进入指定的网址 (localhost:8000//admin) 2. 用户输入用户名和密码 3. 进入管理后台界面
扩展流程	1. 如果用户未进入网址, 则无法操作 2 2. 如果用户未填写或填写错误, 则无法进入 3
后置条件	无

表 2.6.0.2: 管理员登录

用例编号	1.3
用例名称	网站用户音乐播放
执行者	管理员/用户
描述	管理员/用户播放音乐
前置条件	进入网站的音乐播放页面
基本流程	1. 用户进入指定的网址 2. 音乐自动播放 3. 可设置音乐的播放状态 (音量、重新播放)
扩展流程	1. 如果用户未进入网址, 则无法操作 2、3
后置条件	无

表 2.6.0.3: 管理员/用户音乐播放

用例编号	1.4
用例名称	网站用户音乐评论
执行者	管理员/用户
描述	管理员/用户评论音乐
前置条件	进入网站的音乐播放页面
基本流程	1. 用户点击歌曲评论按钮 2. 进入歌曲评论页面 3. 书写评论内容 4. 点击发表按钮
扩展流程	1. 如果用户未进入网址, 则无法操作 2 2. 如果用户操作 3 未完成, 无法进入 4
后置条件	无

表 2.6.0.4: 管理员/用户音乐评论

2.7 非功能性需求

1. 安全性需求

保障系统的安全, 确保各项功能实现分级操作, 做好用户信息的保存, 确保个人信息安全

2. 可靠性需求

要求系统做好关于歌曲和用户的信息管理备份和恢复，系统崩溃率小于 1%

3. 可维护性需求

系统采用 Django 的框架，易于进行功能性的维护，有利于日后功能的拓展，实现高内聚，低耦合，利于与网络服务器数据库进行对接，实现云数据的存储备份。

4. 可测试性需求

系统开发时编写对应的说明测试文档，有利于后期测试人员根据说明文档进行有效测试

5. 易用性需求

界面简介明了，信息扁平化，操作简单，使用清晰的构造界面，使网站设计者，使用者和管理者间能够流畅工作，使需求、设计和开发能够高效衔接。

3 概要设计

3.1 页面设计

3.1.1 网站首页

• 网站首页

网站首页是整个网站的主界面，主要显示网站最新的动态消息以及网站的功能导航。网站动态信息以歌曲的动态为主，如热门下载，热门搜索和新歌推荐等，网站的功能导航是将其它页面的链接展示在首页，方便用户访问浏览。

- 歌曲搜索：位于网页顶端，由文本输入框、搜索框和搜索按钮构成，文本输入框下方是热门搜索歌曲
- 轮播图：以歌曲封面进行轮播，单击图片可进入歌曲播放
- 歌曲分类：位于轮播图左侧，按照歌曲类型进行分类
- 热门歌曲：位于轮播图的右侧，按照歌曲的播放量进行排序
- 新歌推荐：按照歌曲的发行时间进行排序
- 热门搜索：按照歌曲的搜索量进行排序
- 热门下载：按照歌曲的下载量进行排序

3.1.2 歌曲排行榜页面

- 歌曲排行榜：歌曲排行榜按照歌曲的播放量进行排序，用户还可以根据歌曲类型进行自定义筛选。
- 歌曲分类：根据歌曲类型进行歌曲筛选，筛选后的歌曲显示在歌曲列表中
- 歌曲列表：歌曲信息以播放次数降序显示，若对歌曲类型筛选，则对同类型歌曲以播放次数进行降序显示

3.1.3 歌曲播放页面

- 歌曲播放：歌曲播放是为用户提供在线试听功能，此外提供歌曲下载，歌曲点评和相关歌曲推荐
- 歌曲信息：包括歌名、歌手、所属专辑、语种、流派、发行时间、歌词、歌曲封面和歌曲文件等。
- 歌曲下载：提供歌曲下载按钮，供用户进行歌曲的下载，每下载一次歌曲下载次数累加一次
- 歌曲点评：单击歌曲点评按钮进入歌曲点评页面
- 播放列表：记录当前用户试听的记录，每播放一次都会对歌曲的播放次数累计一次
- 相关歌曲：根据当前歌曲类型筛选出同一个类型的其它歌曲的信息，并展示歌曲封面

3.1.4 歌曲点评页面

- 歌曲点评：歌曲点评是通过歌曲播放页进入，每条点评信息包含用户名、点评内容和点评时间
- 由文本输入框和发表按钮组成的表单，以 POST 请求形式实现内容提交
- 点评信息列表：列出当前歌曲的点评信息，并对点评信息设置分页功能

3.1.5 歌曲搜索页面

- 歌曲搜索：歌曲搜索根据用户提供的关键字进行歌曲和歌手匹配查询，搜索结果以数据列表显示在网页上

- 若文本框的内容为空，则默认返回前 50 首最新发行的歌曲
- 若文本框的内容不为空，则从歌曲的歌名或歌手进行匹配查询，查询结果以歌曲的发行时间排序
- 每次搜索时，若文本框的内容与歌名完全相符，则相符的歌曲将其搜索次数累加一次

3.1.6 用户中心页面

- 用户管理：用户管理分为用户注册、登录和用户中心。用户中心包含用户信息、登录注销和歌曲播放记录
- 用户基本信息：显示当前用户的用户头像和用户名，并设有用户退出登录链接
- 歌曲播放记录：播放记录来自于歌曲播放页面的播放列表，并对播放记录进行分页处理
- 用户注册：填写用户名、手机号、用户密码，其中用户名和手机密码具有唯一性，且不为空
- 用户登录：根据用户注册时所填写的手机号或者用户名实现用户登录

3.2 系统架构图

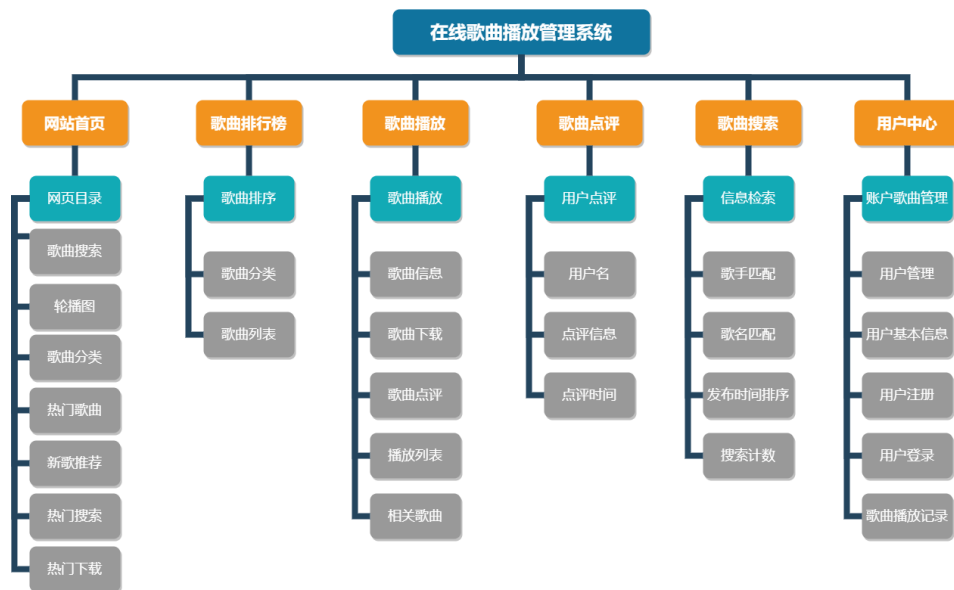


图 3.2.0.1: 在线音乐播放系统架构图

4 详细设计

4.1 软件进度计划

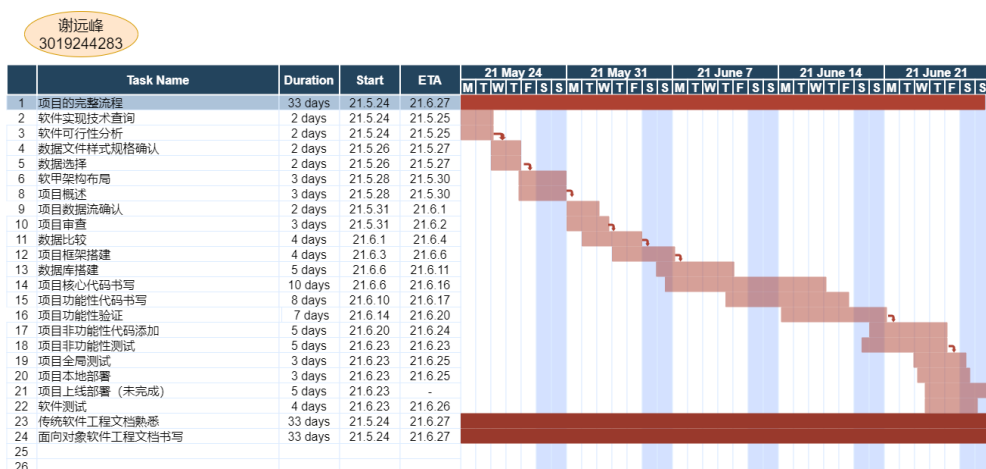


图 4.1.0.1: 软件系统构建计划

5 数据库设计

5.1 数据表

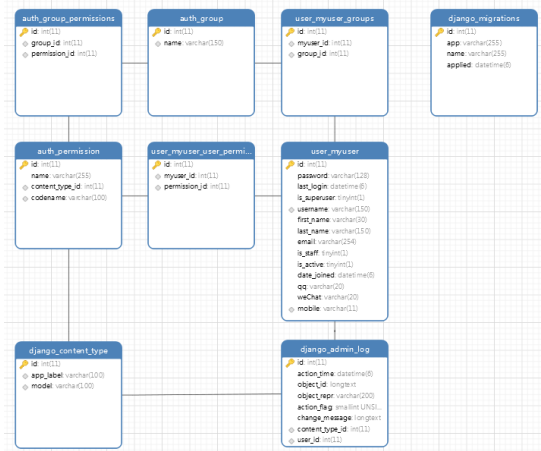


图 5.1.0.1: 数据库表 1

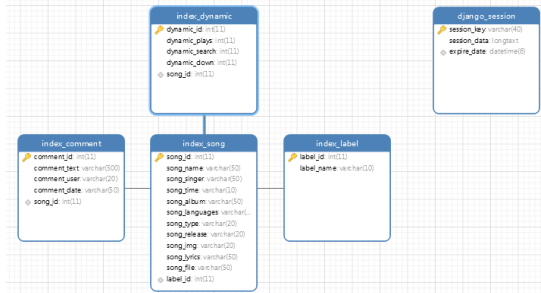


图 5.1.0.2: 数据库表 2

5.2 数据表内容

表字段	字段类型	含义
song_id	Int 类型，长度为 11	主键
song_name	Varchar 类型，长度为 50	歌曲名称
song_singer	Varchar 类型，长度为 50	歌曲演唱歌手
song_time	Varchar 类型，长度为 10	歌曲播放时长
song_album	Varchar 类型，长度为 50	歌曲所属专辑
song_languages	Varchar 类型，长度为 20	歌曲语种
song_type	Varchar 类型，长度为 20	歌曲风格
song_release	Varchar 类型，长度为 20	歌曲发行时间
song_image	Varchar 类型，长度为 20	歌曲封面图片路径
song_lyrics	Varchar 类型，长度为 50	歌曲歌词文件路径
song_file	Varchar 类型，长度为 20	歌曲文件路径
label_id	Int 类型，长度为 11	外键，关联歌曲分类表

表 5.2.0.1: 歌曲信息表 song 的数据结构

表字段	字段类型	含义
dynamic_id	Int 类型，长度为 11	主键
dynamic_plays	Int 类型，长度为 11	歌曲播放次数
dynamic_search	Int 类型，长度为 11	歌曲搜索次数
dynamic_down	Int 类型，长度为 11	歌曲下载次数
song_id	Int 类型，长度为 11	外键，关联歌曲信息表

表 5.2.0.2: 歌曲动态表的数据结构

表字段	字段类型	含义
label_id	Int 类型，长度为 11	主键
label_plays	Varchar 类型，长度为 10	歌曲分类标签

表 5.2.0.3: 歌曲分类表的数据结构

表字段	字段类型	含义
ID	Int 类型，长度为 11	主键
Password	Varchar 类型，长度为 128	用户密码
last_login	Datetime 类型，长度为 6	上次登录时间
is_superuser	Tinyint 类型，长度为 1	超级用户
Username	Varchar 类型，长度为 150	用户名称
first_name	Varchar 类型，长度为 30	用户名字
last_name	Varchar 类型，长度为 150	用户姓氏
Email	Varchar 类型，长度为 254	邮箱地址
is_staff	Tinyint 类型，长度为 1	登录 Admin 权限
is_active	Tinyint 类型，长度为 1	用户的激活状态
date_joined	Datetime 类型，长度为 6	用户创建时间
Qq	Varchar 类型，长度为 20	用户 QQ 账号
weChat	Varchar 类型，长度为 20	用户的微信号码
Mobile	Varchar 类型，长度为 11	用户手机号码

表 5.2.0.4: 用户表的数据结构

表字段	字段类型	含义
comment_id	Int 类型，长度为 11	主键
comment_text	Varchar 类型，长度为 500	歌曲点评内容
comment_user	Varchar 类型，长度为 20	用户名
comment_date	Varchar 类型，长度为 50	点评日期
song_id	Int 类型，长度为 11	外键，关联歌曲信息表

表 5.2.0.5: 歌曲点评表的数据结构

5.3 功能模块设计

5.3.1 网站首页

网站首页以数据查询为主，有 Django 内置的 ROM 框架提供的 API 实现数据查询，查询结果主要以模板语法 for 标签和 if 标签共同实现输出并转换成相应的 HTML 文件

5.3.2 歌曲排行榜页面

歌曲排行榜以 GET 请求进行歌曲的筛选，若不存在请求参数，则将全部歌曲按照播放量进行排序显示，若存在请求参数，则对歌曲进行筛选并按照播放量进行排序显示，歌曲排行榜使用 Django 的通用视图实现

5.3.3 歌曲播放页面

歌曲播放主要实现文件下载、Session 的应用和数据库操作。使用 StreamingHttpResponse 对象作为响应方式，为用户提供文件下载功能；歌曲播放列表使用 Session 实现，主要对 Session 的数据进行读写操作，数据库操作主要对歌曲动态表 dynamic 进行数据的新增和更新

5.3.4 歌曲点评页面

歌曲点评主要使用表单和分页功能。歌曲点评框是由 HTML 编写的表单实现，通过视图函数的参数 request 获取表单数据，实现数据入库处理；分页功能是将当前歌曲点评信息进行分页显示

5.3.5 用户中心页面

用户管理是在 Django 的 Auth 认证系统上实现；用户信息是在内置模型 User 基础上进行扩展；用户注册是在内置表单类 UserCreationForm 的基础上实现；用户登录由内置函数 check_password 和 login 共同实现；用户中心使用过滤器 login_required 实现访问限制，并由处理器 context_processor_auth 自动生成用户信息，最后使用 Session 和分页功能实现歌曲播放记录的显示。

5.3.6 网站后台

网站后台主要使用 Admin 后台的基本配置，如 App 的命名方法，Admin 的标题设置和模型注册与设置。App 的命名方法是由 App 的初始化文件 __init__.py 实现的，Admin 的标题设置和模型注册与设置在 App 的 admin.py 中实现

6 系统部署

6.1 部署环境

- 硬件环境
应用数据库：Mysql, 数据库文件管理：Navicat for Mysql, Intel Core i5 8th 1T 硬盘，16G 内存
- 软件环境
编辑环境：Vscode, 后端：Anaconda 3.8.5 64bit + Django2.2.14
- 网络环境
校园网本机环境 localhost/127.0.0.1
- 使用说明
python 采用 anaconda 的虚拟配置环境，运行文件需修改静态文件目录 (music/music/settings.py 141 行)

6.2 代码浏览

```
17 from django.contrib import admin
18 from django.urls import path, include
19 from django.conf.urls import url
20 from django.views import static
21 from django.conf import settings
22
23 urlpatterns = [
24     path('admin/', admin.site.urls),
25     path('', include('index.urls')),
26     path('ranking.html', include('ranking.urls')),
27     path('play/', include('play.urls')),
28     path('comment/', include('comment.urls')),
29     path('search/', include('search.urls')),
30     path('user/', include('user.urls')),
31     # 设置项目上线的静态资源路径
32     url('^static/(?P<path>.*)$', static.serve,
33         {'document_root': settings.STATIC_ROOT}, name='static'),
34 ]
35
36
37 # 设置404、500错误状态码
38 handler404 = views.page_not_found
39 # handler500 = views.page_not_found_error
```

图 6.2.0.1: 软件系统构建计划

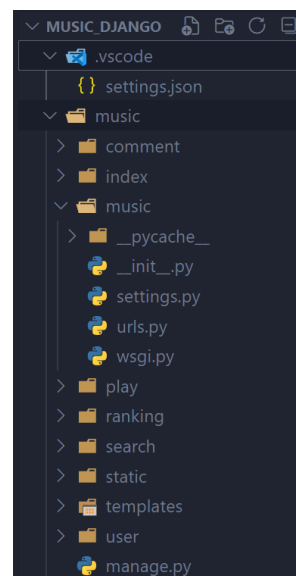


图 6.2.0.2: 软件系统构建计划

7 软件测试

7.1 单元用例测试

7.1.1 黑盒单元测试

输入条件	有效等价类	编号	无效等价类	编号
用户名	用户名未被注册	1-1	用户名未填写	1-6
			用户名已被注册	1-7
手机号	11 位数字	1-2	手机号未填写	1-8
			包含非法字符	1-9
密码	4-16 位字符	1-3	未输入密码	1-10
			小于 4 个字符	1-11
			大于 16 个字符	1-12
	英文、数字、特殊字符组合	1-4	未全部包含	1-13
确认密码	重复密码	1-5	未输入重复密码	1-14
			重复错误	1-15

表 7.1.1.1: 注册等价类划分

输入条件	有效等价类	编号	无效等价类	编号
用户名	用户名称	2-1	用户名未填写	2-4
	用户手机号	2-2	用户名不存在	2-5
密码	与账号对应	2-3	密码未填写	2-6
			密码错误	2-7

表 7.1.1.2: 登录等价类划分

输入条件	有效等价类	编号	无效等价类	编号
歌名	歌曲名称	3-1	歌名未填写	3-11
歌手	歌手名称	3-2	歌手未填写	3-12
时长	播放时长	3-3	包含异常字符	3-13
			时间未填写	3-14
专辑	专辑名称	3-4	密码错误	3-15
语种	语言	3-5	语种未填写	3-16
类型	类型	3-6	类型未填写	3-17
发行时间	时间	3-7	时间格式错误	3-18
歌曲图片	图片路径	3-8	路径不存在	3-19
歌词	歌词路径	3-9	路径不存在	3-20
歌曲分类	歌曲分类	3-10	分类不存在	3-21
			分类未填写	3-22

表 7.1.1.3: 歌曲信息添加修改等价类划分

输入条件	有效等价类	编号	无效等价类	编号
歌名	歌曲名称	4-1	歌名未填写	4-5
播放次数	正整数	4-2	负数	4-6
搜索次数	正整数	4-3	负数	4-7
下载次数	正整数	4-4	负数	4-8

表 7.1.1.4: 歌曲动态修改添加等价类划分

输入条件	有效等价类	编号	无效等价类	编号
内容	评论内容	5-1	文本过长	5-5
用户名称	用户名称	5-2	未输入	5-6
歌名	歌名	5-3	无效歌名	5-7
日期	时间格式	5-4	非法样式	5-8

表 7.1.1.5: 歌曲评论添加删除等价类划分

7.1.2 白盒单元测试

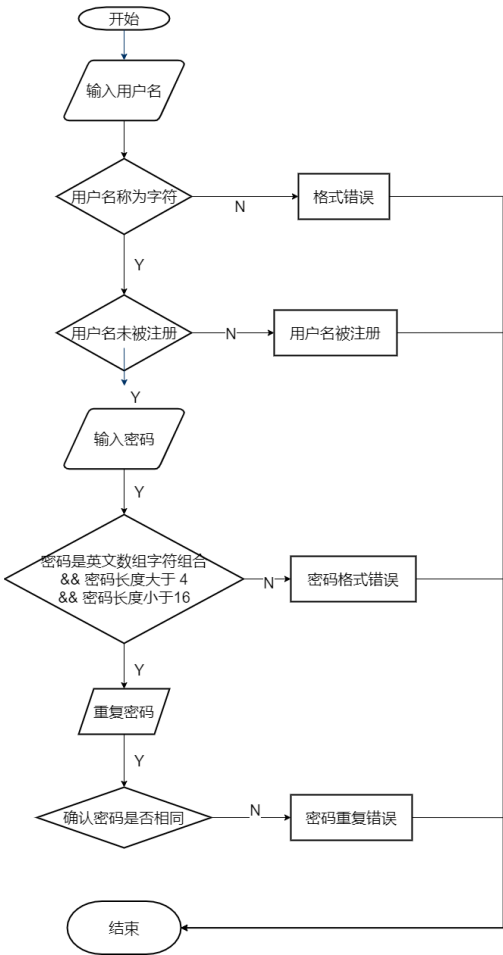


图 7.1.2.1: 注册信息流程图

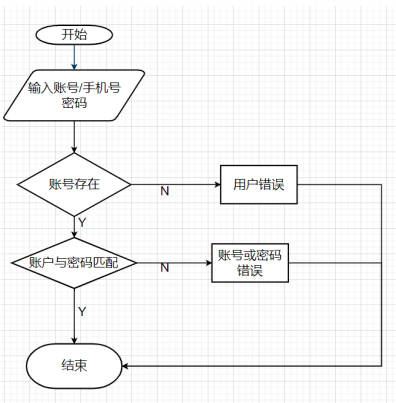


图 7.1.2.2: 登录信息流程图

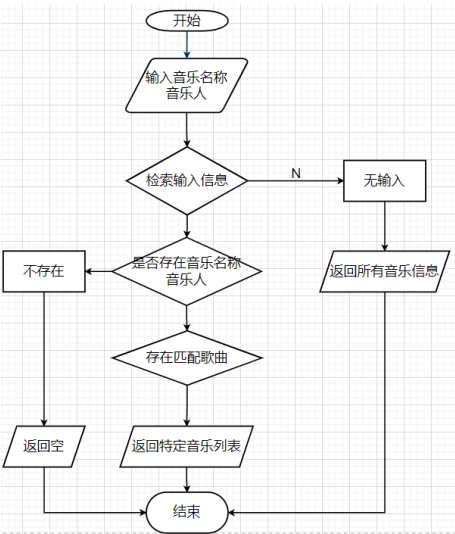


图 7.1.2.3: 检索音乐流程图

用户名	密码	确认密码	预期成果
XYF	1234qwer	1234qwer	注册成功
XYF	1234qwer	dsadb	注册失败
XYF	1234343	1234343	密码格式错误
XYF	13345r	13345rd	注册失败
admin	1234qwer	1234qwer	用户名已经注册
XYF	13232vfdvfv3f3rf34ff	13232vfdvfv3f3rf34ff	密码过长
XYF	2we	2we	密码过短

表 7.1.2.1: 注册信息流程表

账号存在状态	密码匹配状态	预期成果
XYF	1234qwer	登录成功
XYF1	1234qwer	账户错误
XYF	1234qwer1	密码错误
不存在	不匹配	用户名不存在
不存在	不匹配	用户名不存在

表 7.1.2.2: 登录信息流程表

检索名称	对应歌手或歌曲是否存在	预期结果
无	存在	检索全部
歌曲信息	存在	输出列表
歌曲信息	不存在	输出空
歌手信息	存在	输出列表
歌手信息	不存在	输出空

表 7.1.2.3: 检索信息流程表

7.2 测试截图

7.2.1 注册功能测试截图



图 7.2.1.2: 注册信息 2

图 7.2.1.1: 注册信息 1

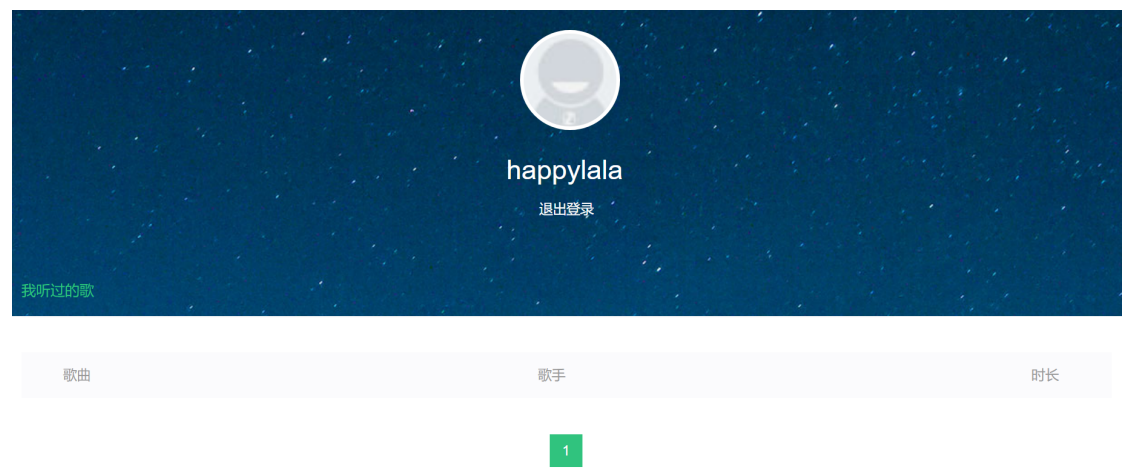


图 7.2.1.3: 注册信息 2

7.2.2 登录模块测试截图

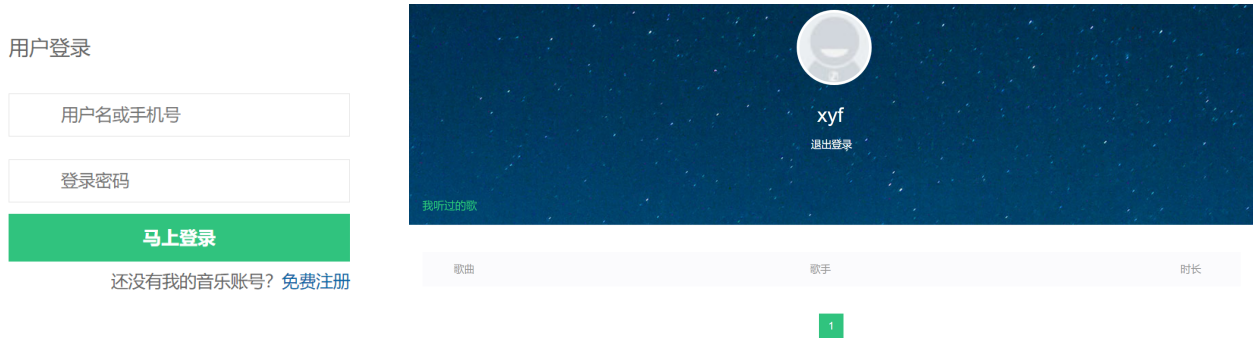


图 7.2.2.1: 登录记录 1

图 7.2.2.2: 登录记录 2



图 7.2.2.3: 网站首页

8 项目总结

本项目为在线歌曲播放系统，用户可以在线收听歌曲，了解并收听较新发布时间的歌曲，查看不同主题分类的歌曲，根据用户的收听量和搜索量进行歌曲的热度排序系统管理员可以对用户进行账户信息的增删查改和歌曲信息的增删查改

通过本次项目,完整体会并了解软件开发的全过程,了解并尝试面向对象工程的开发流程理解并应用 Django 的框架进行在线歌曲播放网站的开发，对软件内部的架构有了更加深刻的了解