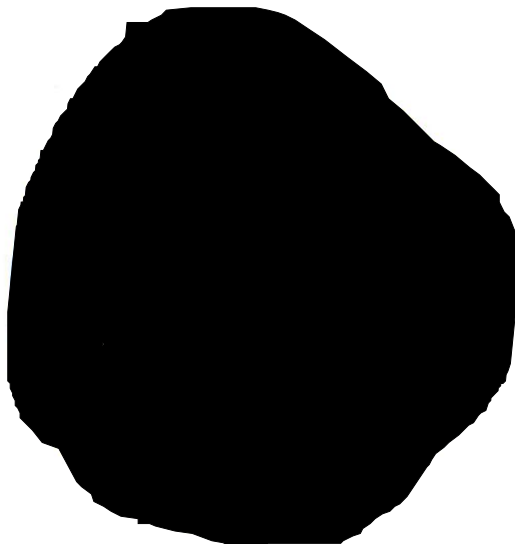


数据库系统原理

音乐游戏曲库管理系统开发报告



学号： [REDACTED]

姓名： [REDACTED]

专业： [REDACTED]

授课老师： [REDACTED]

目录

- 一、问题定义3
 - 1.1 背景介绍3
 - 1.2 编写目的3
 - 1.3 核心功能概述.....3
- 二、可行性分析4
 - 2.1 技术上的可行性4
 - 2.2 管理上的可行性4
 - 2.3 安全上的可行性4
- 三、需求分析4
 - 3.1 功能性需求4
 - 3.2 数据字典8
- 四、总体设计11
 - 4.1 数据库设计11
 - 4.2 应用系统设计18
- 五、详细设计19
 - 5.1 数据库设计19
 - 5.2 应用系统设计27
- 六、功能实现30
 - 6.1 数据库建立30
 - 6.2 客户端应用系统实现32
 - 6.3 后端系统实现.....51
- 七、软件测试52
 - 7.1 数据库安全评估52
 - 7.2 端口检测53
 - 7.3 漏洞检测53
- 八、系统运行维护54

一、问题定义

1.1 背景介绍

音乐游戏 (Music Game): 用动作 (键盘、控制器、身体动作等) 与节奏吻合的游戏。音乐游戏不断出现的各种按键 (NOTE) 合成一首歌曲, 类似于奏乐。在音乐游戏 (Music Game) 中, 玩家配合音乐与节奏做出动作 (依画面指示按钮、踏舞步、操作模仿乐器的控制器等) 来进行游戏。通常玩家做出的动作与节奏吻合即可增加得分, 相反情况下则会扣分。这类游戏主要考的是玩家对节奏的把握, 以及手指的灵活性, 个人的反应和眼力。所以音乐游戏是一种健康绿色的游戏。一款音乐游戏的基本构成主要是**音游曲**和**音游谱面**。

音游曲: 通常是有较强节奏感的音乐, 便于玩家随着音乐律动做出动作。通常由曲师编曲制作。音游曲通常还会搭配**音游 BGA**。

音游谱面: 由多个音符组成的音符集合, 根据音乐的节奏有序地在画面上出现、移动和消失, 玩家根据节奏在适当的时机在画面上与该音符做交互 (例如点击、长按、滑动等), 游戏系统对玩家交互的时机与音乐节奏的重合程度, 给出相应得分。由专门的**谱师**制作。

音游 BGA: BGA (background animation), 指与音游曲目配套的一段视频, 传达该音乐的意境, 由专门的**BGA 制作师**完成。

THE BMS OF FIGHTERS (BOF): 通俗来说, 是一个音游曲目的比赛, 创作者组成队伍, 队伍主要由曲师、谱师和 BGA 制作师构成, 共同完成一整套的音游曲作品。最终参与投票的玩家会根据音游曲、音游谱面和音游 BGA 三者的综合质量投票, 并根据每个作品的投票数进行排名, 决出冠亚季军。最终, BOF 比赛中的优秀作品会作为一些音乐游戏的曲目被收录。

1.2 编写目的

大部分关于音乐游戏的知识库和百科信息都较为分散, 不利于玩家直观地查看相关信息。本数据库项目旨在为玩家提供一个曲库管理系统, 查看与音游、音游曲、曲师、谱师、BOF 相关的信息, 收藏自己喜欢的曲目。

本篇实验报告主要就此系统进行前期的需求分析、概念设计和逻辑设计。

1.3 核心功能概述

系统需要实现的主要功能:

- √ 建立数据库: 根据系统功能设计的要求以及用户需求, 设计相关数据库。
- √ 音游信息管理: 实现音乐游戏的基本信息的录入、删除操作, 对音乐游戏的基本信息进行查询。
- √ 作品信息管理: 对音游相关作品 (音游曲、音游谱面、BGA) 的相关的内容进行录入、删除等操作, 并允许玩家对作品进行查询。
- √ 创作者信息管理: 对音游创作者 (曲师、谱师、BGA 制作师) 的基本信息进行录入、删除等操作, 并允许玩家对创作者的信息和作品进行查询。
- √ 音游比赛 (BOF) 信息管理: 对音游比赛的信息进行录入和删除操作, 并允许玩家对音

游比赛的详细信息进行查询。

二、可行性分析

2.1 技术上的可行性

该系统使用 C/S 架构，客户端使用 Android 应用实现，服务器端上部署数据库。

- 服务端：
Win10 云服务器 + Tomcat + Eclipse EE + MySQL
开发语言：Java
- 客户端
Win10 + Android Studio + SQLite
开发语言：Java

安卓 APP 的使用较为广泛，比较有利于推广使用，同时有很多可用的接口，易于开发设计；服务端使用较为经典的数据库环境，兼容性较强，并便于维护。

2.2 管理上的可行性

在后端，系统管理员可以在服务器端对数据库进行修改，并上传音乐游戏的 bga，使前端的用户可以查看数据库信息。

在前端，管理员可以在 APP 上登录管理员账号，对用户系统进行管理，进行删除和添加的操作。

2.3 安全上的可行性

采用 C/S 架构，安全性比较强。

在建立数据库时可以通过对数据库的设计用户名和密码进行保密，以及在建立数据库以后可以对数据进行压缩等技术，保证数据的安全，使数据库具有安全保障。

三、需求分析

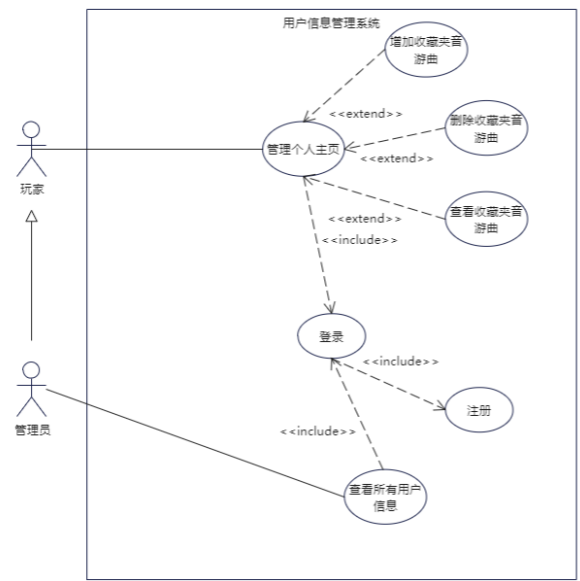
3.1 功能性需求

3.1.1 系统用例概述

用户信息管理系统	
用例	用例概述
管理用户个人信息	用户注册登录后，可以修改自己的个人信息，如 ID、头像、性别、生日等。

增加/删除/查看收藏夹中的音游歌曲	用户可以进入自己的收藏夹，并查看、删除自己收藏过的音游歌曲，也可以对喜欢的歌曲进行收藏操作。
-------------------	--

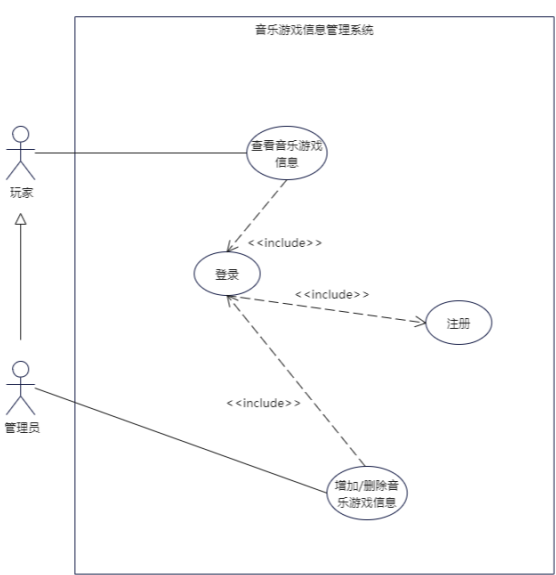
用例图：



音乐游戏信息管理系统

用例	用例概述
增加/删除音乐游戏信息（管理员）	上传关于音乐游戏的信息，如游戏名、上架时间、收录歌曲等。并可以对其进行删除。
查看音乐游戏信息	用户可以根据查看想要了解的音乐游戏的相关信息。

用例图：

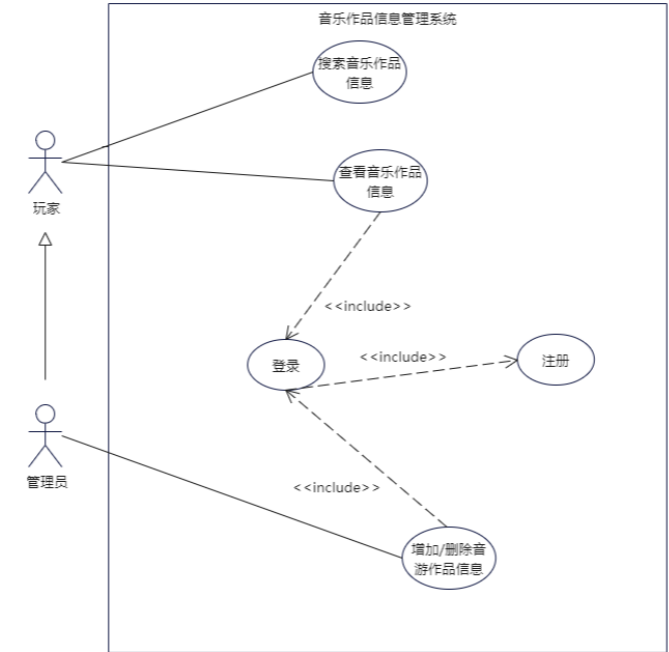


音游作品管理系统

用例	用例概述
增加/删除音游作品信息（管理员）	上传关于音游曲目的信息，如曲名、创作者、将音游作品收录的音游等。并可以对其进行删除。

搜索作品	用户可以对想要了解的作品进行搜索，获得结果。
查看作品	用户可以查看作品并进行播放等操作（可能通过链接形式，可能通过直接播放）

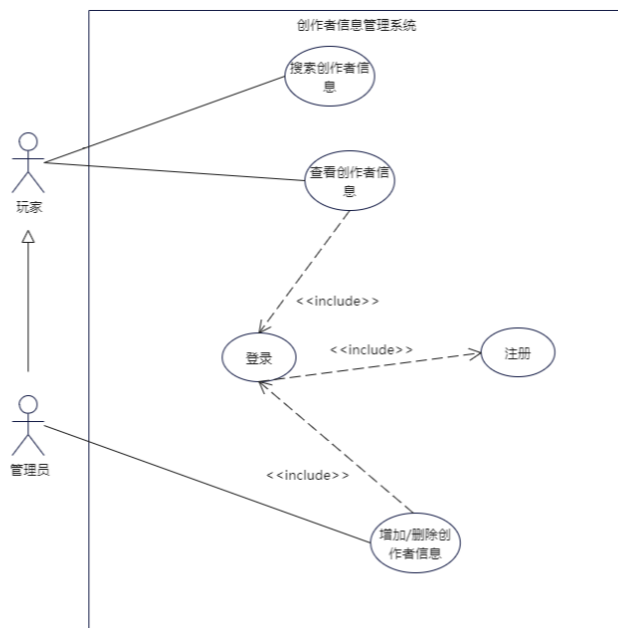
用例图：



创作者信息管理系统

用例	用例概述
增加/删除创作者信息（管理员）	上传创作者的信息，如创作者的名字、简介等。
搜索创作者信息	用户可以对想要了解的创作者进行搜索，获得结果。
查看创作者信息	用户可以查看创作者信息。

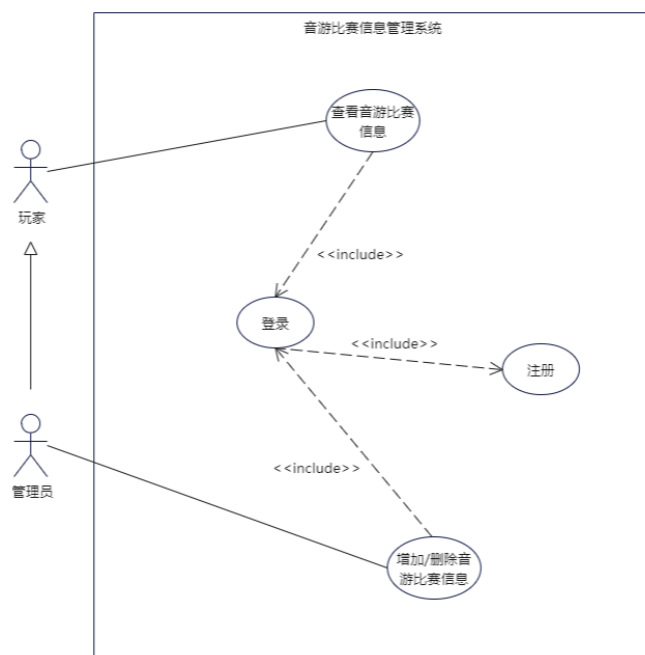
用例图：



音游比赛管理系统

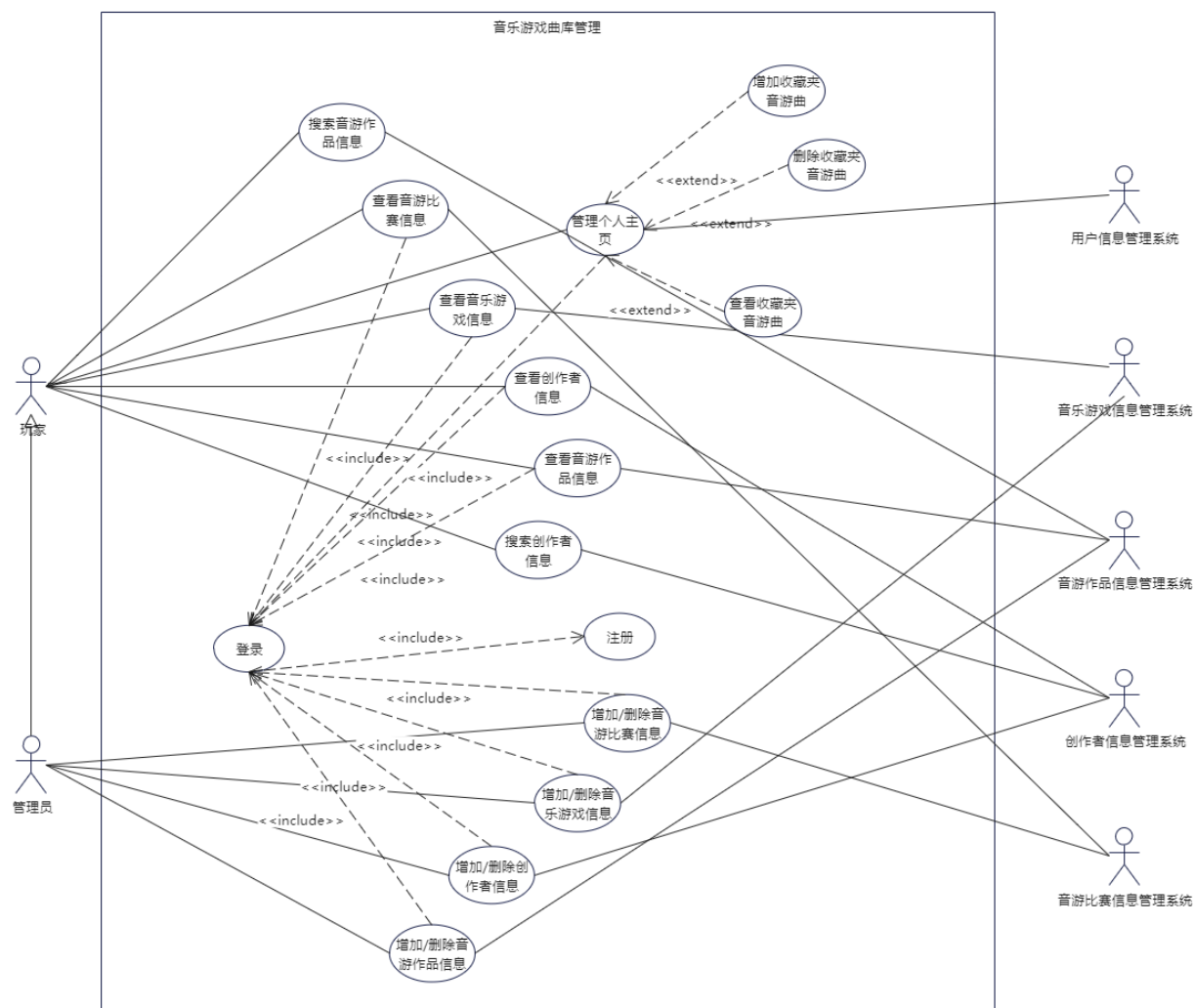
用例	用例概述
增加/删除音游比赛信息（管理员）	上传关于音游比赛的信息，如年份、介绍、冠亚季军信息等。
查看音游比赛信息	用户可以查询与音游比赛相关的信息。

用例图：



3.1.2 总体用例

总体系统用例图



3.2 数据字典

3.2.1 user_info

记录玩家/用户的个人信息。

属性名	字段	类型	约束
用户 ID	user_id	int	主键, not null
用户密码	user_password	varchar	not null
用户昵称	user_nickname	varchar	not null
用户性别	user_sex	varchar	'f', 'm' or null
用户生日	user_birthday	datetime	

3.2.2 admin

记录管理员的账号信息。

属性名	字段	类型	约束
管理员 ID	admin_id	int	主键, not null
管理员密码	admin_password	varchar	

3.2.3 game_info

记录音乐游戏的基础信息。

属性名	字段	类型	约束
游戏名	game_name	varchar	主键, not null
游戏别名	game_cname	varchar	
发行日期	game_time	date	
发行公司	game_company	varchar	
游戏平台	game_platform	varchar	
游戏简介	game_intro	varchar	

3.2.4 music_info

记录音乐游戏收录的音乐曲目的基本信息。

属性名	字段	类型	约束
曲目 ID	music_id	int	主键, not null
曲目名称	music_name	varchar	
创作曲师	music_creator	varchar	
音乐链接	music_link	varchar	

3.2.5 beatmap_info

记录音乐游戏收录的音乐曲目对应谱面的基本信息。

属性名	字段	类型	约束
谱面 ID	beatmap_id	int	主键, not null
创作谱师	beatmap_creator	varchar	
谱面链接	beatmap_link	varchar	

3.2.6 bga_info

记录音乐游戏收录的音乐曲目对应 BGA 的基本信息。

属性名	字段	类型	约束
BGA 的 ID	bga_id	int	主键, not null
bga 的作者	bga_creator	varchar	
bga 链接	bga_link	varchar	

3.2.7 bof

记录音游比赛的基础信息。

属性名	字段	类型	约束
bof 比赛的名称	bof_name	varchar	主键, not null
冠军曲 ID	first_id	varchar	not null
亚军曲 ID	second_id	varchar	not null
季军曲 ID	third_id	varchar	not null
比赛官方链接	bof_link	varchar	

3.2.8 include

记录音游和音游曲目的收录关系。

属性名	字段	类型	约束
音游曲的 ID	include_id	int	主键, not null
收录音游曲的游戏名	include_game	varchar	主键, not null
该游戏中的最低难度等级	include_level1	int	
该游戏中的中等难度等级	include_level2	int	
该游戏中的较高难度等级	include_level3	int	
该游戏中的最高难度等级	include_level4	int	

3.2.9 team

BOF 比赛的参赛队伍。

属性名	字段	类型	约束
参赛的作品 ID	team_music_id	int	主键, not null
bof 比赛的名称	team_bof_name	varchar	
参赛的队伍名称	team_name	varchar	
参赛获得的名次	team_place	varchar	

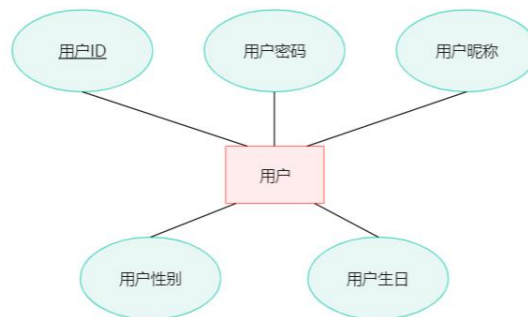
四、总体设计

4.1 数据库设计

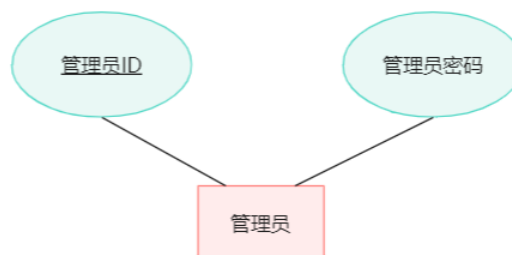
数据库的总体设计可以通过实体 E-R 图进行体现。

4.1.1 实体属性设计

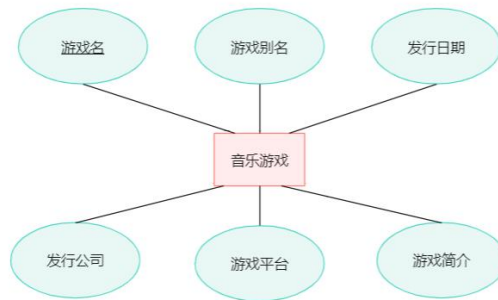
4.1.1.1 user 属性设计



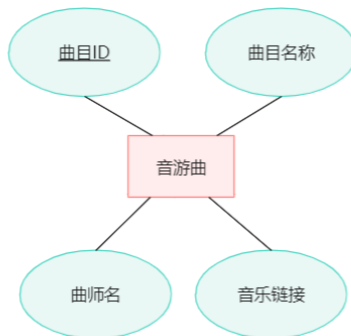
4.1.1.2 admin 属性设计



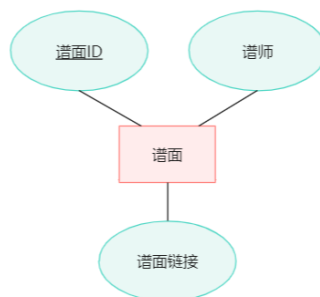
4.1.1.3 game_info 属性设计



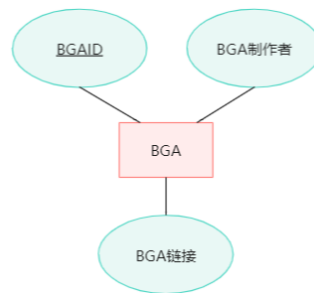
4.1.1.4 music_info 属性设计



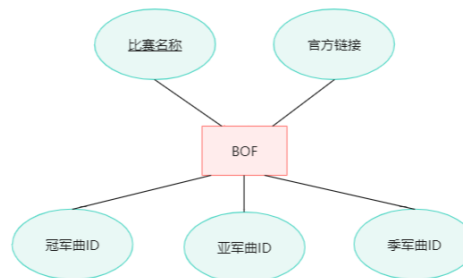
4.1.1.5 beatmap_info 属性设计



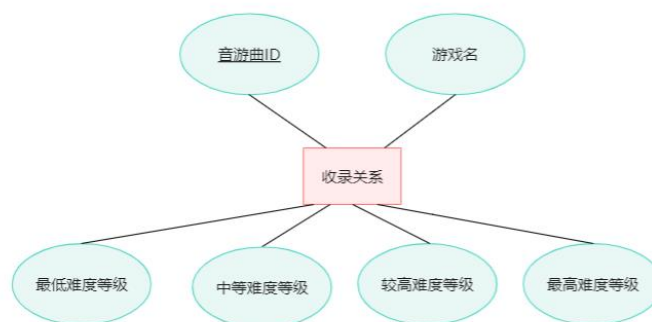
4.1.1.6 bga_info 属性设计



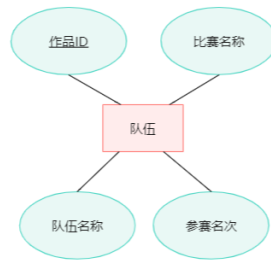
4.1.1.7 bof 属性设计



4.1.1.8 include 属性设计

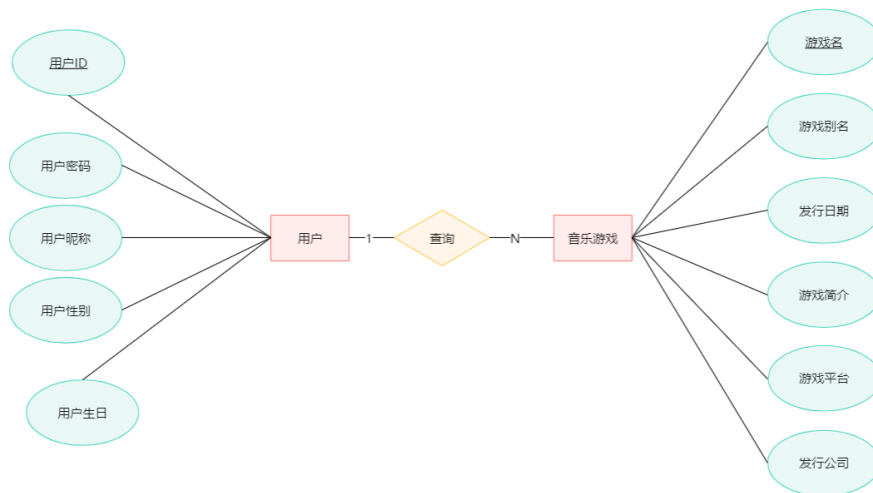


4.1.1.9 team 属性设计

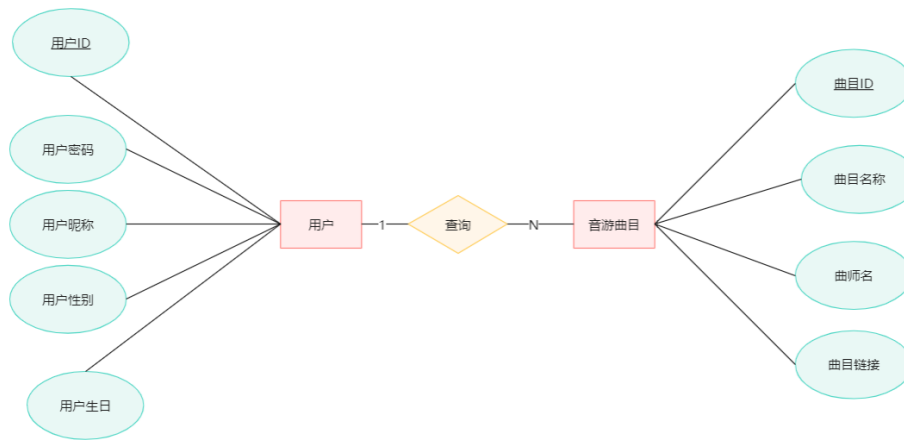


4.1.2 实体关系设计

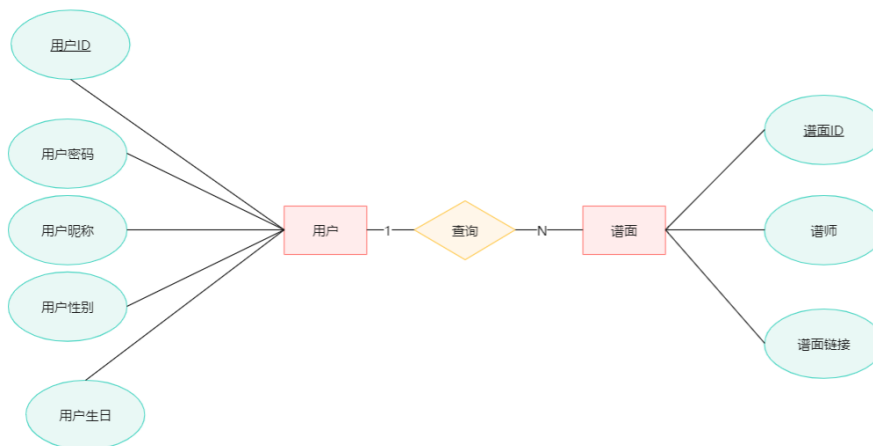
4.1.2.1 用户和音乐游戏关系



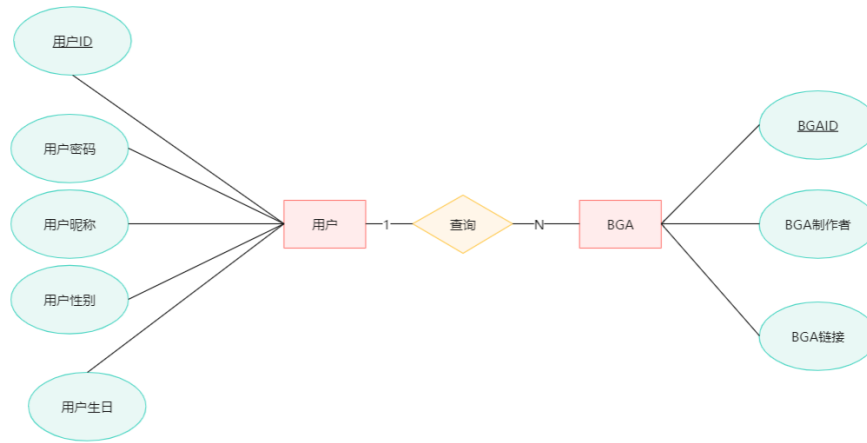
4.1.2.2 用户和音游曲关系



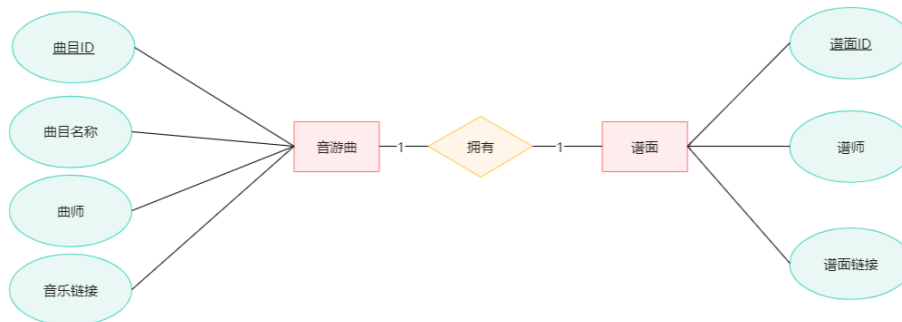
4.1.2.3 用户和谱面关系



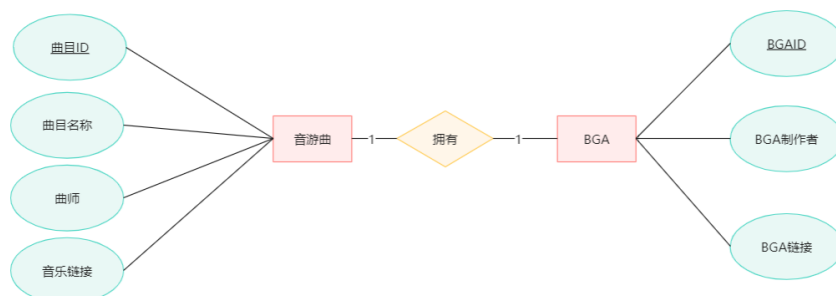
4.1.2.4 用户和 BGA 关系



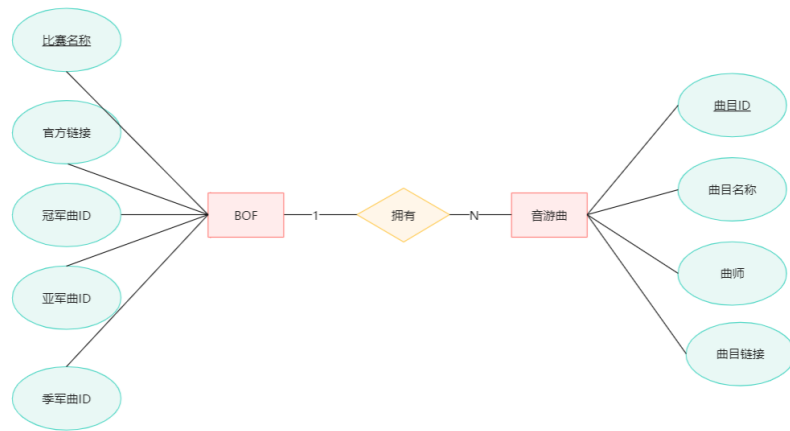
4.1.2.5 音游曲和谱面的关系



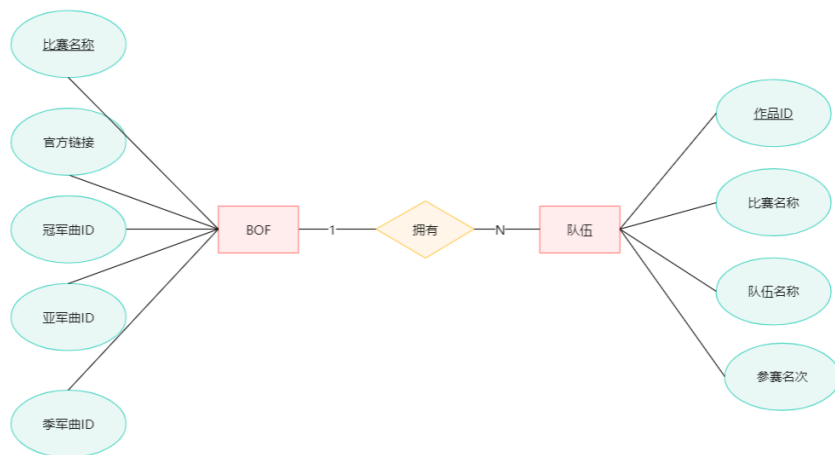
4.1.2.6 音游曲和 BGA 的关系



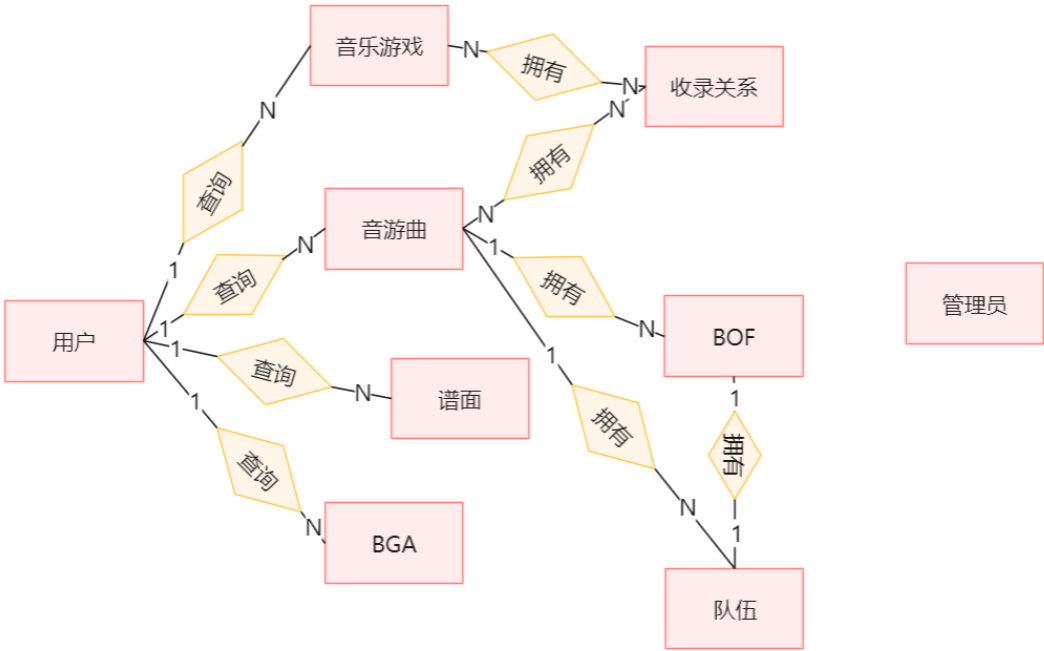
4.1.2.7 音游曲和音游比赛的关系



4.1.2.8 音游比赛 BOF 和队伍的关系



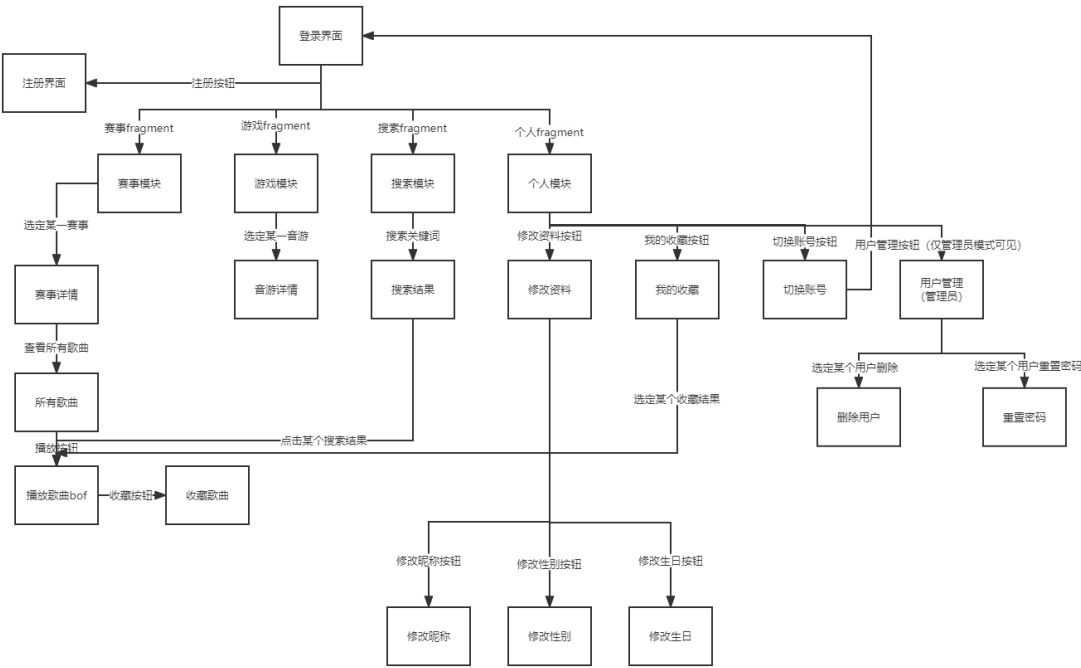
4.1.3 全局 E-R 图



4.2 应用系统设计

根据以上的数据库设计，对该安卓 APP 进行系统应用的总体设计。根据功能划分，将功能划分为多个模块，每个模块对应数据库的某些数据库表。

界面关系如下图所示。



由图可见，本安卓应用的界面关系如图所示，由四个 fragment 组成的四个模块展开，在后面将对每个界面的功能进行详细的描述。

五、详细设计

5.1 数据库设计

5.1.1 概念设计

5.1.1.1 用户信息表

实体说明

用户信息表登记了玩家/用户的 ID、密码等用户的基本信息。

实体设计

user_info	
PK	user_id
	user_password
	user_nickname
	user_sex
	user_birthday

属性说明

√ user_id：用户（玩家）的 ID 值，作为该表的主码，是一个数字，用于唯一标识用户的身份。

√ user_password：用户密码，用于用户登录时身份的验证，是用户身份证明的不可缺少的属性，非空。

√ user_nickname：用户昵称，用户可以根据个人喜好，设置自己的昵称，非空。

√ user_sex：用户的性别，分为男和女，用户也可以选择不设置。

√ user_birthday：用户的生日。

5.1.1.2 管理员信息表

实体说明

管理员信息表登记了管理员的 ID 和密码信息。

实体设计

admin_info	
PK	admin_id
	admin_password

属性说明

√ admin_id：管理员的 ID 值，作为该表的主码，是一个数字，用于唯一标识管理员的身份。

√ admin_password：管理员的密码，用于管理员登录时身份的验证，是管理员身份证明的不

可缺少的属性，非空。

5.1.1.3 音乐游戏信息表

实体说明

音乐游戏信息表登记了音乐游戏的名字、别名、发行日期等基本信息，用于客户查询了解。

实体设计

game_info	
PK	game_name
	game_cname
	game_time
	game_company
	game_platform
	game_intro

属性说明

- √ game_name：音乐游戏的名字，作为该表的主码，作为唯一区分该游戏的标识。
- √ game_cname：音乐游戏的别名，有的音乐游戏可能会存在的其他版本的名称。
- √ game_time：音乐游戏的发行时间。
- √ game_company：音乐游戏的发行公司。
- √ game_platform：该音乐游戏支持的平台（PC、Android、iOS 等）。
- √ game_intro：对该音乐游戏的概括性描述。

5.1.1.4 音游曲目信息表

实体说明

音游曲目信息表记录的是那些用于作为音游中的音乐的曲目信息。

实体设计

music_info	
PK	music_id
	music_name
	music_creator
	music_link

属性说明

- √ music_id：曲目的名称，用于唯一标识音游曲目。是音游曲目表的主键。
- √ music_name：曲目的名称，由创作的曲师设计的曲名。
- √ music_creator：曲师的称呼，是创作该音乐的制作人。
- √ music_link：该音乐对应的链接。

5.1.1.5 谱面信息表

实体说明

谱面信息表上记录的是某个曲目所搭配的谱面的具体信息, 由于不同游戏的音游谱面不尽相同, 这里统一采用参加音游竞赛的官方谱面。

实体设计

beatmap_info	
PK	beatmap_id
	beatmap_creator
	beatmap_link

属性说明

- √ beatmap_id: 谱面的 ID, 用于唯一标识一份谱面。作为谱面信息表的主键, 同时作为外码, 和音游曲的 ID 相关联。
- √ beatmap_creator: 谱师的称呼, 即制作该谱面的制作人。
- √ beatmap_link: 该谱面的对应链接。

5.1.1.6 音游 BGA 信息表

实体说明

音游 BGA 信息表主要记录了有关音乐游戏曲目配套动画 (BGA) 的基本信息。

实体设计

bga_info	
PK	bga_id
	bga_creator
	bga_link

属性说明

- √ bga_id: BGA 的 ID, 用于唯一标识一份 BGA。作为音游 BGA 信息表的主键, 同时作为外码, 和音游曲的 ID 相关联。
- √ bga_creator: BGA 制作人的称呼, 即制作该 BGA 的制作人。
- √ bga_link: 该 BGA 的对应链接。

5.1.1.7 音游比赛 BOF 信息表

实体说明

音游比赛 BOF 信息表记录了音游体系中较为有名的比赛 BMS OF FIGHTERS (BOF) 的基本信息。

实体设计

bof	
PK	bof_name
	first_id
	second_id
	third_id
	bof_link

属性说明

- √ bof_name: 对于每一年, 音游比赛 BOF 的比赛名称会有细微差别, 因此用名称来区分每

- 一年的 BOF 比赛。是该表的主键
- √ first_id: 获得冠军的曲目 ID。
 - √ second_id: 获得亚军的曲目 ID。
 - √ third_id: 获得季军的曲目 ID。
 - √ bof_link: 当年 BOF 比赛的官方链接。

5.1.1.8 音游收录关系表

实体说明

音游收录关系表记录了音游曲目在音游中收录的情况，收录涉及到玩家游玩，所以还有与该曲目在该游戏中难度的设置。

实体设计

include	
PK	include_id
PK	include_game
	include_level1
	include_level2
	include_level3
	include_level4

属性说明

- √ include_id: 被音游收录的曲目的 ID。和该表中的游戏字段共同构成该表的主键。同时作为外码与音游曲目的 ID 相关联。
- √ include_game: 收录该音游曲目的音乐游戏。和该表中的曲目 ID 字段共同构成该表的主键。
- √ include_level1: 对于游戏中的最低难度，在游戏中标明的等级。
- √ include_level2: 对于游戏中的中等难度，在游戏中标明的等级。
- √ include_level3: 对于游戏中的较高难度，在游戏中标明的等级。
- √ include_level4: 对于游戏中的最高难度，在游戏中标明的等级。如果该游戏只有三个难度阶级，则这个字段为空。

5.1.1.9 参赛队伍信息表

实体说明

参赛队伍信息表记录了参与音游比赛 BOF 的各个队伍的信息。每个队伍由三个组成：曲师、谱师和 BGA 制作人，这个关系会和曲目表、谱面表和 BGA 信息表相关联。

实体设计

team	
PK	team_music_id
	team_bof_name
	team_name
	team_place

属性说明

- √ team_music_id: 参赛队伍共同完成的作品 ID，作为参赛队伍信息表的主键。
- √ team_bof_name: 参赛队伍参加的比赛名称，即参加的哪一期比赛。
- √ team_name: 参赛队伍的名称。

√ team_place: 参赛队伍所创作的作品在该次比赛中获得的名次。

5.1.2 逻辑设计

5.1.2.1 依赖关系描述与优化

1. 用户信息表

根据依赖关系，一旦用户的账号 ID 被确定，那么用户的密码、昵称、性别和生日也被唯一确定了，因此有依赖关系：

用户 ID→密码，用户 ID→昵称，用户 ID→性别，用户 ID→生日

主码为用户 ID，关系模式满足 3NF，无需再分解。

关系模式有：

$F = \{user_id, user_password, user_nickname, user_sex, user_birthday\}$

$U = \{user_id \rightarrow user_password, user_id \rightarrow user_nickname, user_id \rightarrow user_sex, user_id \rightarrow user_birthday\}$

2. 管理员信息表

根据依赖关系，一旦管理员的账号 ID 被确定，那么管理员的密码也被唯一确定，因此有依赖关系：

管理员 ID→密码

主码为管理员 ID，关系模式满足 3NF，无需再分解。

关系模式有：

$F = \{admin_id, admin_password\}$

$U = \{admin_id \rightarrow admin_password\}$

3. 音乐游戏信息表

根据依赖关系，一旦确定了音乐游戏的名字，别名、发行日期、发行公司、游戏平台 and 游戏简介等信息也会被唯一确定，因此有依赖关系：

游戏名→游戏别名，游戏名→发行日期，游戏名→发行公司，游戏名→游戏平台，游戏名→游戏简介

主码为游戏名，关系模式满足 3NF，无需再分解。

关系模式有：

$F = \{game_name, game_cname, game_time, game_company, game_platform, game_intro\}$

$U = \{game_name \rightarrow game_cname, game_name \rightarrow game_time, game_name \rightarrow game_company, game_name \rightarrow game_platform, game_name \rightarrow game_intro\}$

4. 音游曲目信息表

根据依赖关系，一旦曲目的 ID 被唯一确定，曲目名称、作曲师和音乐链接就也被唯一确定，因此有依赖关系：

曲目 ID→曲目名称，曲目 ID→作曲师，曲目 ID→音乐链接

主码为曲目 ID，关系模式满足 3NF，无需再分解。

关系模式有：

$U = \{music_id, music_name, music_creator, music_link\}$

$F = \{music_id \rightarrow music_name, music_id \rightarrow music_creator, music_id \rightarrow music_link\}$

5. 谱面信息表

根据依赖关系，一旦谱面的 ID 被唯一确定，创作谱师、谱面链接也就被唯一确定，因此有依赖关系：

谱面 ID→创作谱师，谱面 ID→谱面链接

主码为谱面 ID，关系模式满足 3NF，无需再分解。

关系模式有：

$U = \{\text{beatmap_id}, \text{beatmap_creator}, \text{beatmap_link}\}$

$F = \{\text{beatmap_id} \rightarrow \text{beatmap_creator}, \text{beatmap_id} \rightarrow \text{beatmap_link}\}$

6. BGA 信息表

根据依赖关系，一旦 BGA 的 ID 被唯一确定，BGA 的制作者、BGA 链接也就被唯一确定，因此有依赖关系：

BGA 的 ID→BGA 制作者，BGA 的 ID→BGA 链接

主码为用户 ID，关系模式满足 3NF，无需再分解。

关系模式有：

$U = \{\text{bga_id}, \text{bga_creator}, \text{bga_link}\}$

$F = \{\text{bga_id} \rightarrow \text{bga_creator}, \text{bga_id} \rightarrow \text{bga_link}\}$

7. 音游比赛 BOF 信息表

根据依赖关系，一旦音游比赛的名称被唯一确定，冠军曲的 ID、亚军曲的 ID、季军曲的 ID 和比赛官方链接也就被唯一确定，因此有依赖关系：

比赛名称→冠军曲 ID，比赛名称→亚军曲 ID，比赛名称→季军曲 ID，比赛名称→比赛官方链接

主码为比赛名称，关系模式满足 3NF，无需再分解。

关系模式有：

$U = \{\text{bof_name}, \text{first_id}, \text{second_id}, \text{third_id}, \text{bof_link}\}$

$F = \{\text{bof_name} \rightarrow \text{first_id}, \text{bof_name} \rightarrow \text{second_id}, \text{bof_name} \rightarrow \text{third_id}, \text{bof_name} \rightarrow \text{bof_link}\}$

8. 曲目收录表

根据依赖关系，一旦音游曲的 ID 和收录该曲子的音游被唯一确定，收录的各级难度就被唯一确定，因此有依赖关系：

(音游曲 ID，游戏名)→最低难度等级，(音游曲 ID，游戏名)→中等难度等级，(音游曲 ID，游戏名)→较高难度等级，(音游曲 ID，游戏名)→最高难度等级

主码为音游曲 ID 和游戏名，关系模式满足 3NF，无需再分解。

关系模式：

$U = \{\text{include_id}, \text{include_game}, \text{include_level1}, \text{include_level2}, \text{include_level3}, \text{include_level4}\}$

$F = \{(\text{include_id}, \text{include_game}) \rightarrow \text{include_level1}, (\text{include_id}, \text{include_game}) \rightarrow \text{include_level2}, (\text{include_id}, \text{include_game}) \rightarrow \text{include_level3}, (\text{include_id}, \text{include_game}) \rightarrow \text{include_level4}\}$

9. 参赛队伍信息表

根据依赖关系，一旦参赛作品 ID 被唯一确定，音游比赛名称、参赛队伍名称、获得名次也唯一确定，因此有依赖关系：

参赛作品 ID→音游比赛名称，参赛作品 ID→参赛队伍名称，参赛作品 ID→参赛名次

主码为参赛作品 ID，关系模式满足 3NF，无需再分解。

$U = \{\text{team_music_id}, \text{team_bof_name}, \text{team_name}, \text{team_place}\}$

$F = \{\text{team_music_id} \rightarrow \text{team_bof_name}, \text{team_music_id} \rightarrow \text{team_name}, \text{team_music_id} \rightarrow \text{team_place}\}$

5.1.2.2 表结构设计

1. user_info

属性名	字段	类型	约束
用户 ID	user_id	int	主键, not null
用户密码	user_password	varchar(20)	not null
用户昵称	user_nickname	varchar(20)	not null
用户性别	user_sex	varchar(5)	'f', 'm' or null
用户生日	user_birthday	date	——

2. admin

属性名	字段	类型	约束
管理员 ID	admin_id	int	主键, not null
管理员密码	admin_password	varchar(20)	not null

3. game_info

属性名	字段	类型	约束
游戏名	game_name	varchar(20)	主键, not null
游戏别名	game_cname	varchar(20)	——
发行日期	game_time	date	——
发行公司	game_company	varchar(20)	——
游戏平台	game_platform	varchar(20)	——
游戏简介	game_intro	varchar(20)	——

4. music_info

属性名	字段	类型	约束
曲目 ID	music_id	int	主键/外键, not null
曲目名称	music_name	varchar(20)	not null
作曲师	music_creator	varchar(20)	——
音乐链接	music_link	varchar(20)	——

5. beatmap_info

属性名	字段	类型	约束
谱面 ID	beatmap_id	int	主键/外键, not null
创作谱师	beatmap_creator	varchar(20)	——
谱面链接	beatmap_link	varchar(20)	——

6. bga_info

属性名	字段	类型	约束
BGA 的 ID	bga_id	int	主键/外键, not null
bga 的制作者	bga_creator	varchar(20)	——
bga 链接	bga_link	varchar(20)	——

7. bof

属性名	字段	类型	约束
bof 比赛的名称	bof_name	varchar(20)	主键, not null
冠军曲 ID	first_id	int	外键, not null
亚军曲 ID	second_id	int	外键, not null

季军曲 ID	third_id	int	外键, not null
比赛官方链接	bof_link	varchar(20)	——

8. include

属性名	字段	类型	约束
音游曲的 ID	include_id	int	主键/外键, not null
收录音游曲的游戏名	include_game	varchar(20)	主键/外键, not null
该游戏中的最低难度等级	include_level1	int	——
该游戏中的中等难度等级	include_level2	int	——
该游戏中的较高难度等级	include_level3	int	——
该游戏中的最高难度等级	include_level4	int	——

9. team

属性名	字段	类型	约束
参赛的作品 ID	team_music_id	int	主键, not null
bof 比赛的名称	team_bof_name	varchar(20)	外键, not null
参赛的队伍名称	team_name	varchar(20)	——
参赛获得的名次	team_place	varchar(20)	——

5.1.3 物理设计

在进行了概念设计和逻辑设计以后, 需要对物理设计进行分析。在前端的界面和后端的数据库进行数据交互的时候, 涉及到数据的频繁访问, 需要对经常受到访问的数据进行索引的设计。

5.1.3.1 功能描述

1. 用户注册/登录

该音乐游戏曲库管理平台需要用户 (玩家) 进行注册与登录的操作来实现对信息的查询功能, 需要对用户的账号信息进行匹配, 对密码进行验证等操作。

由于用户的 ID 是主码, 是不重复的, 因此对用户 ID 的索引可以设置为唯一索引。因此, 用户的 ID 可以按照数字排序并进行唯一性查找。

2. 有限信息查询

有一些数量较为有限的数据, 比如音游比赛的信息、音乐游戏的信息, 由于音游比赛、音乐游戏的数据量较小, 因此不需要花费太多的时间和功耗就可以达到及时查询的效果, 因此对这类有限信息的查询, 就采用不建立索引的方式, 就可以达到基础的查询需求。

3. 大量信息查询

有一些数量较为庞大的数据, 比如音游曲目、音游谱面、音游 BGA、参赛队伍等信

息，由于这些数据数量大，因此需要花费很多时间，需要高效一些的索引方式，便于实现查询功能。

因此相比直接使用字符串，而是又建立了一个 int 类型的索引，并用整数的方式进行表之间的连接和查询过程。

5.1.3.2 索引设计

结合上面所分析的过程，对数据库中的索引进行设计，如下所示：

表名	字段名	查询类型	索引类型
用户信息表	用户 ID	equal	排序-二分-唯一
音游曲目表	音游曲 ID	equal	主索引
音游谱面表	谱面 ID	equal	主索引
音游 BGA 表	BGAID	equal	主索引

5.2 应用系统设计

下面将对本应用系统的各大功能进行阐述。

5.2.1 登录功能设计

登录功能涉及本安卓应用的登陆界面和服务端端的登陆相关 servlet，并与数据库中的 user 和 admin 表相关。

登录界面可以切换用户和管理员模式，对 APP 进行登录。其中包含昵称和密码的输入，一旦用户的输入符合服务端端的数据库中的信息，则可以登陆成功，使用 APP。

5.2.2 注册功能设计

注册功能涉及本安卓应用的注册界面和服务端端的注册相关 servlet，并与数据库中的 user 表相关。

注册界面可以供用户进行注册账号的操作。其中包括昵称的输入、密码的输入和密码的确认。该功能仅限于注册普通用户，而对管理员来说是无效的。

5.2.3 赛事功能设计

赛事功能涉及本安卓应用的赛事界面和服务端端的赛事相关 servlet，并与数据库中的 bof 表相关。

赛事功能将对近几年来 BOF 音游比赛进行一一列举，通过点击详情，就可以查看赛事相关的信息。

5.2.4 赛事详情功能设计

赛事详情功能涉及本安卓应用的赛事详情界面和服务端端的赛事信息相关 servlet，并与数据库中的 bof 表和 music 表相关。

赛事详情功能对某一年 BOF 赛事进行信息介绍，主要是前三名的获奖作品，并提供按钮以查看所有的歌曲和排名。

5.2.5 参赛曲目功能设计

参赛曲目功能涉及本安卓应用的参赛曲目界面和服务端端的参赛曲目相关 servlet，并与数据库中的 music 表和 bof 表相关。

参赛曲目功能列举出该年参赛的所有作品，并提供每首作品的播放 BOF 的 bga 的通道，用户可以通过此通道来欣赏曲目的 BGA。

5.2.6 播放 BGA 功能设计

播放 BGA 功能涉及本安卓应用的播放 BGA 界面和服务端中存储的 BGA 视频。

该功能可以播放选定的 BOF 曲目，并可以选择是否收藏该曲目，便于在个人界面的收藏部分进行查看。

5.2.7 游戏展示功能设计

游戏展示功能涉及本安卓应用的游戏列表展示页面和服务端端的游戏展示相关 servlet，并与数据库中的 game 表相关。

该功能展示所有音乐游戏的列表，点击列表当中的游戏表项，可以跳转到游戏信息界面，查看更加详细的该游戏的信息。

5.2.8 游戏信息功能设计

游戏信息功能涉及本安卓应用的游戏信息页面和服务端端的游戏信息相关 servlet，并于数据库中的 game 表相关。

该功能会展示某一个游戏的具体信息，包括游戏的名称、别名、发行日期、发行公司、支持平台和游戏简介等等。并提供查看收录曲目及其对应难度的表格的通道。

5.2.9 游戏收录展示功能设计

游戏收录展示功能涉及本安卓应用的游戏收录展示界面和服务端端的游戏收录展示相关 servlet，并与数据库中的 include 表相关。

该功能会展示游戏中的曲目收录相关信息，包括游戏中曲目的各个难度等级的具体难度，

以表格的形式展现。

5.2.10 搜索曲目功能设计

搜索曲目功能涉及本安卓应用的搜索曲目界面和服务端端的搜索曲目相关的 servlet, 并与数据库中的 music 表相关。

该功能可以输入关键词并搜索, 对数据库中的曲目进行匹配。对匹配到的搜索结果进行点击, 就可以跳转到相应的播放界面, 播放 BGA 视频。

5.2.11 个人中心功能设计

个人中心功能涉及本安卓应用的个人中心界面和服务端端的个人信息相关的 servlet, 并与数据库中的 user 表和 admin 表相关。

该功能将在界面上展示昵称、性别、生日等用户的信息, 为普通用户提供修改资料、我的收藏、切换账号的功能, 并为管理员提供用户管理的按钮, 用于对用户账号进行操作。

5.2.12 修改资料功能设计

修改资料功能涉及本安卓应用的修改资料界面和服务端端的修改个人信息相关的 servlet, 并与数据库中的 user 表和 admin 表相关。

该功能可以对用户的资料进行修改, 包括昵称、性别和生日等。

5.2.13 我的收藏功能设计

我的收藏功能涉及本安卓应用的我的收藏界面和服务端端的获取收藏曲目相关的 servlet, 并与数据库中的 like 表相关。

该功能可以查看用户收藏的曲目。用户点击某一个曲目, 就可以播放该曲目的 BGA。

5.2.14 切换账号功能设计

切换账号功能可以直接退出当前帐号, 然后跳转到登陆界面, 可以借此重新登录另外一个账号了。

5.2.15 用户管理功能设计

用户管理功能仅限于管理员模式, 普通用户无法使用。

用户管理功能涉及本安卓应用的用户管理界面和服务端端的用户管理相关的 servlet, 并与数据库中的 user 表相关。

该功能可以对普通用户进行直接管理, 即删除用户和重置密码的功能, 在前端删除的同

时，对后端的数据库相关表进行修改。


六、功能实现

基于以上的设计，对数据库系统应用进行实现。

6.1 数据库建立

基于以上对数据库的概念设计、逻辑设计和物理设计，对数据库进行建立，如下所示。

1. music_info

Result Grid						
Filter Rows:		Export:  Wrap				
	Field	Type	Null	Key	Default	Extra
▶	music_id	int	NO	PRI	NULL	
	music_name	varchar(255)	NO		NULL	
	music_creator	varchar(255)	YES		NULL	
	music_link	varchar(255)	YES		NULL	

2. user_info

	Field	Type	Null	Key	Default	Extra
▶	user_id	int	NO	PRI	NULL	auto_increment
	user_password	varchar(255)	YES		NULL	
	user_nickname	varchar(255)	YES		NULL	
	user_sex	varchar(5)	YES		NULL	
	user_birthday	date	YES		NULL	

3. admin_info

	Field	Type	Null	Key	Default	Extra
▶	admin_id	int	NO	PRI	NULL	
	admin_password	varchar(255)	YES		NULL	

4. game_info

	Field	Type	Null	Key	Default	Extra
▶	game_name	varchar(255)	NO	PRI	NULL	
	game_cname	varchar(255)	YES		NULL	
	game_time	date	YES		NULL	
	game_company	varchar(255)	YES		NULL	
	game_platform	varchar(255)	YES		NULL	
	game_intro	varchar(255)	YES		NULL	

5. music_info

	Field	Type	Null	Key	Default	Extra
▶	music_id	int	NO	PRI	NULL	
	music_name	varchar(255)	NO		NULL	
	music_creator	varchar(255)	YES		NULL	
	music_link	varchar(255)	YES		NULL	

6. beatmap_info

	Field	Type	Null	Key	Default	Extra
▶	beatmap_id	int	NO	PRI	NULL	
	beatmap_creator	varchar(255)	YES		NULL	
	beatmap_link	varchar(255)	YES		NULL	

7. bga_info

	Field	Type	Null	Key	Default	Extra
▶	bga_id	int	NO	PRI	NULL	
	bga_creator	varchar(255)	YES		NULL	
	bga_link	varchar(255)	YES		NULL	

8. bof_info

	Field	Type	Null	Key	Default	Extra
▶	bof_id	int	YES		NULL	
	bof_name	varchar(20)	NO	PRI	NULL	
	first_id	int	NO		NULL	
	second_id	int	NO		NULL	
	third_id	int	NO		NULL	
	bof_link	varchar(255)	YES		NULL	

9. include

	Field	Type	Null	Key	Default	Extra
▶	include_id	int	NO	PRI	NULL	
	include_game	varchar(255)	NO	PRI	NULL	
	include_level1	float	YES		NULL	
	include_level2	float	YES		NULL	
	include_level3	float	YES		NULL	
	include_level4	float	YES		NULL	
	include_flag	int	YES		NULL	

10. team_info

	Field	Type	Null	Key	Default	Extra
▶	team_music_id	int	NO	PRI	NULL	
	team_bof_name	varchar(20)	YES		NULL	
	team_name	varchar(255)	YES		NULL	
	team_place	int	YES		NULL	

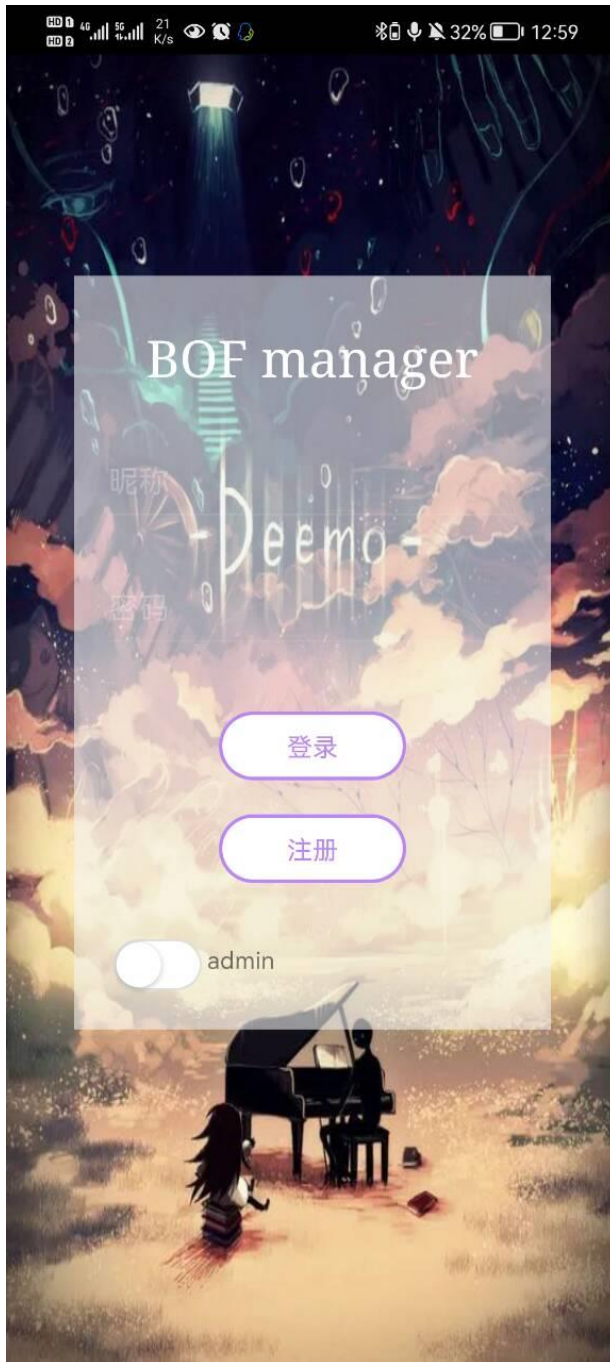
11. like_info

	Field	Type	Null	Key	Default	Extra
▶	user_id	int	NO	PRI	NULL	
	music_id	int	NO	PRI	NULL	

每张数据库表都指定了合适的主键、外键，并设置好约束，对级联都进行了符合要求的设置。

6.2 客户端应用系统实现

6.2.1 登录界面实现

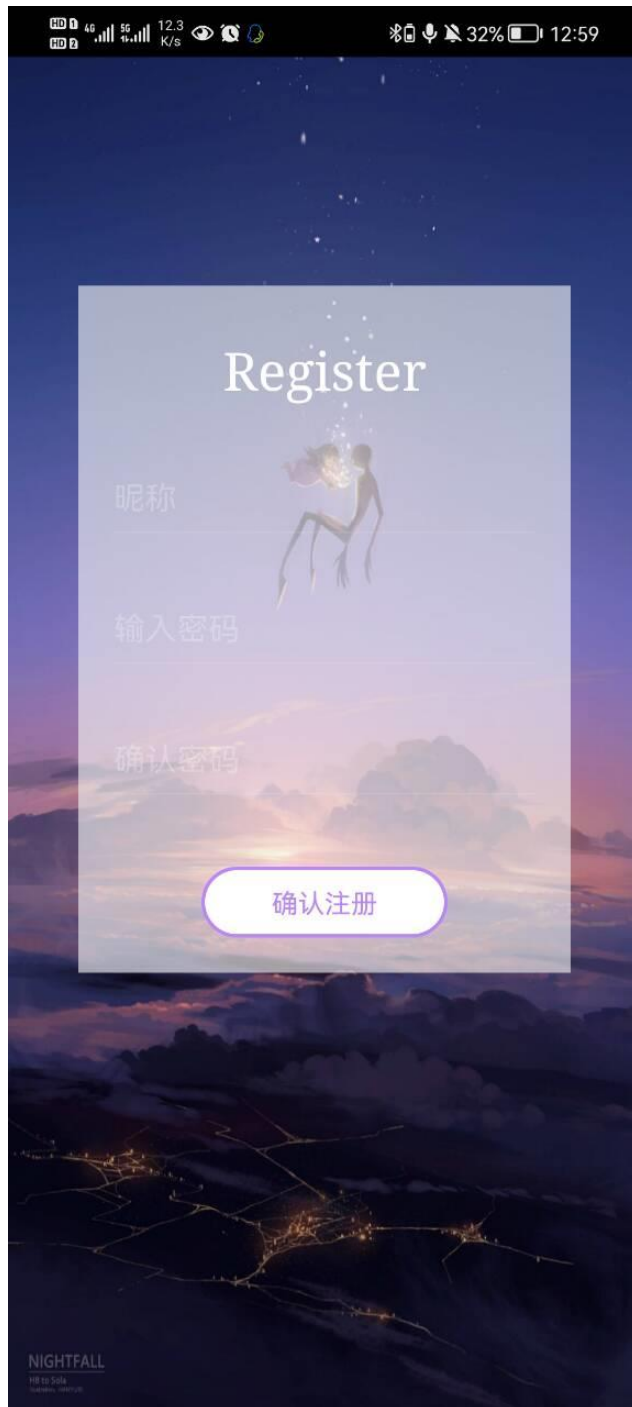


当前为登陆界面。登录界面可以使用左下角的切换按钮选择用户和管理员模式，对APP进行登录。

其中包含昵称和密码的输入框，若两者都输入正确，点击“登录”按钮，则可以登陆成功，使用APP；若输入错误，则会弹出错误提示。

若点击“注册”按钮，则会进入注册界面。

6.2.2 注册界面实现



当前为注册界面。注册界面可以供用户进行注册账号的操作。其中包括昵称的输入、密码的输入和密码的确认。该功能仅限于注册普通用户，而无法注册管理员账号。

5.2.3 赛事界面实现



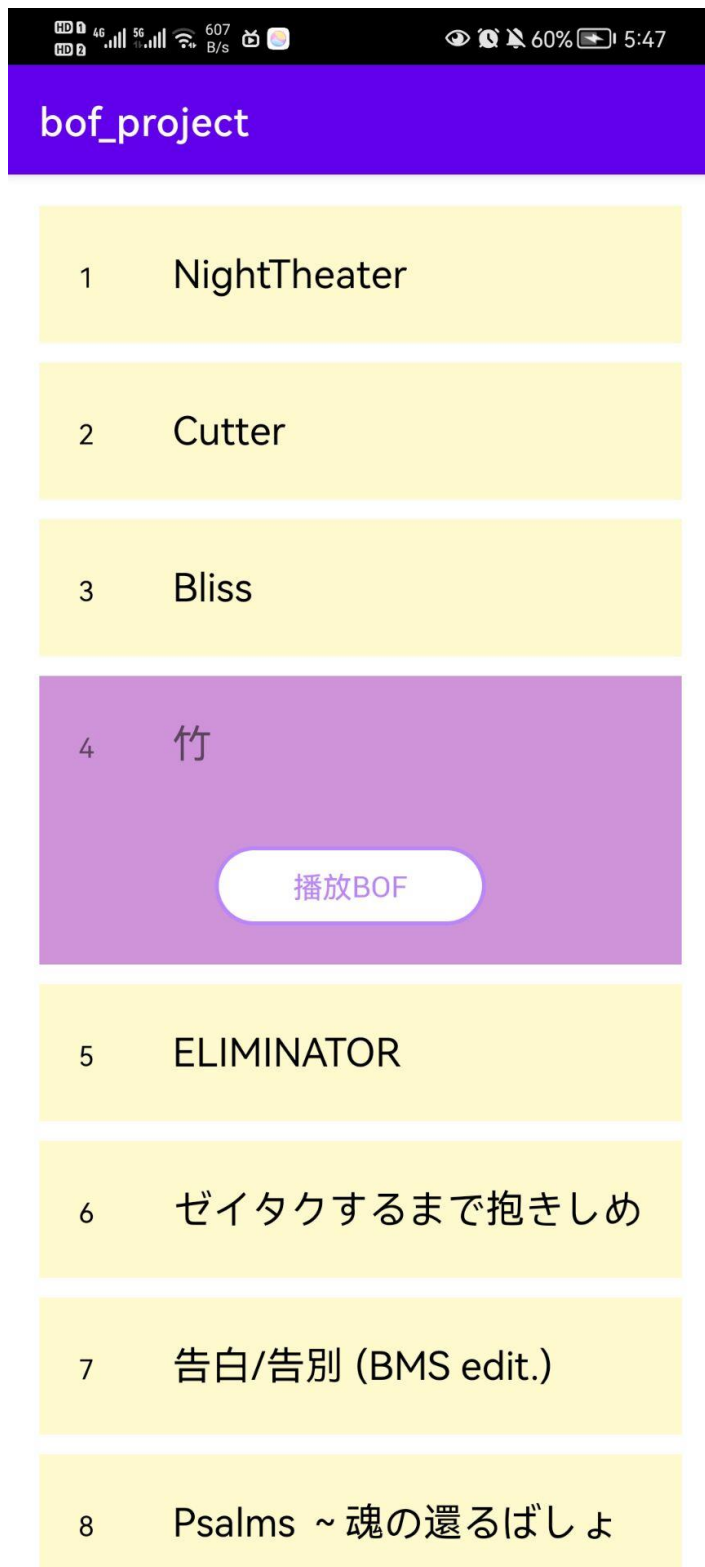
当前为赛事界面。赛事功能将对近几年来 BOF 音游比赛进行一一列举，通过点击详情，就可以查看赛事相关的信息。可以选中其中一个对其查看，点击右侧上下按钮可以切换前一个和后一个比赛，点击收起就会收起所有卡片。点击“查看详情”按钮就可以查看该年音游比赛的具体信息。

5.2.4 赛事详情界面实现



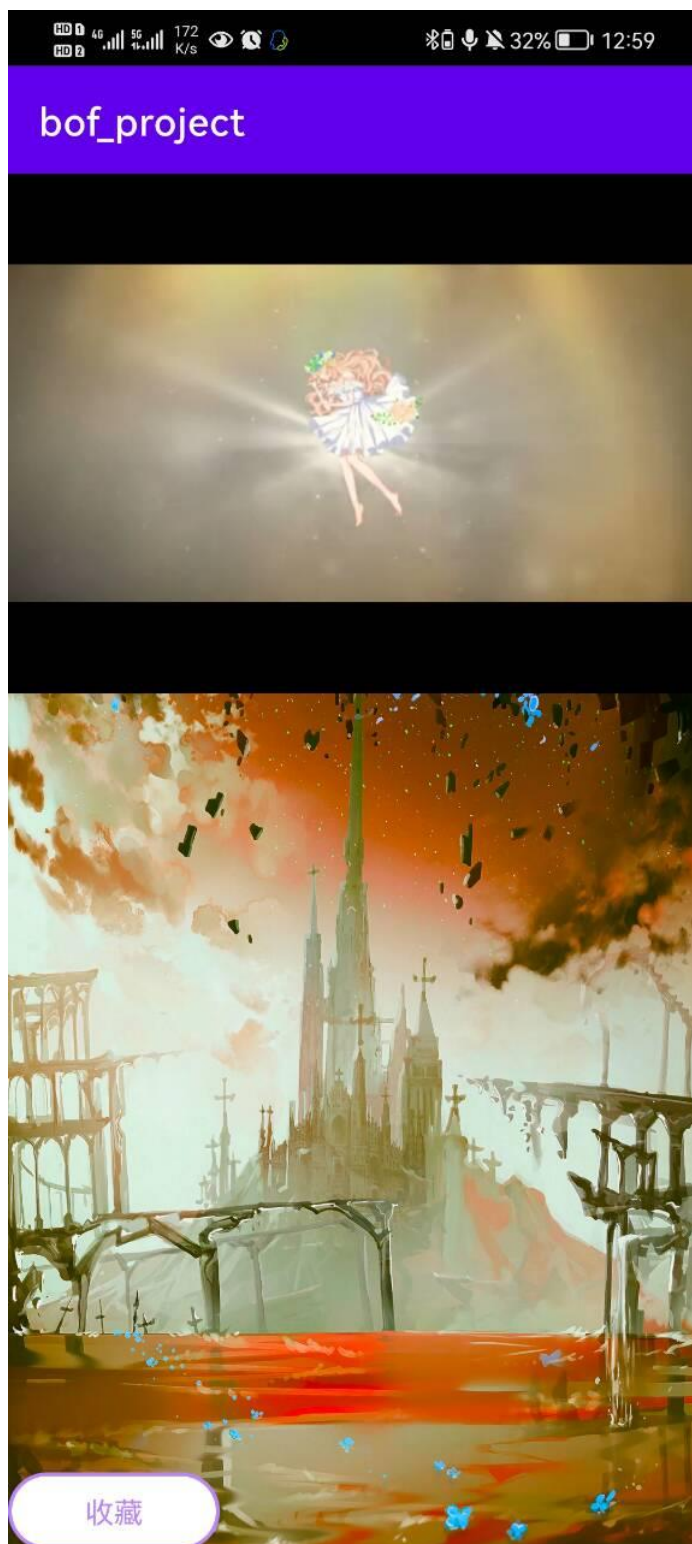
当前为赛事详情界面。赛事详情功能对某一年 BOF 赛事进行信息介绍，主要是前三名的获奖作品，并提供“查看所有歌曲和排名”按钮以查看所有的歌曲和排名。

5.2.5 参赛曲目界面实现



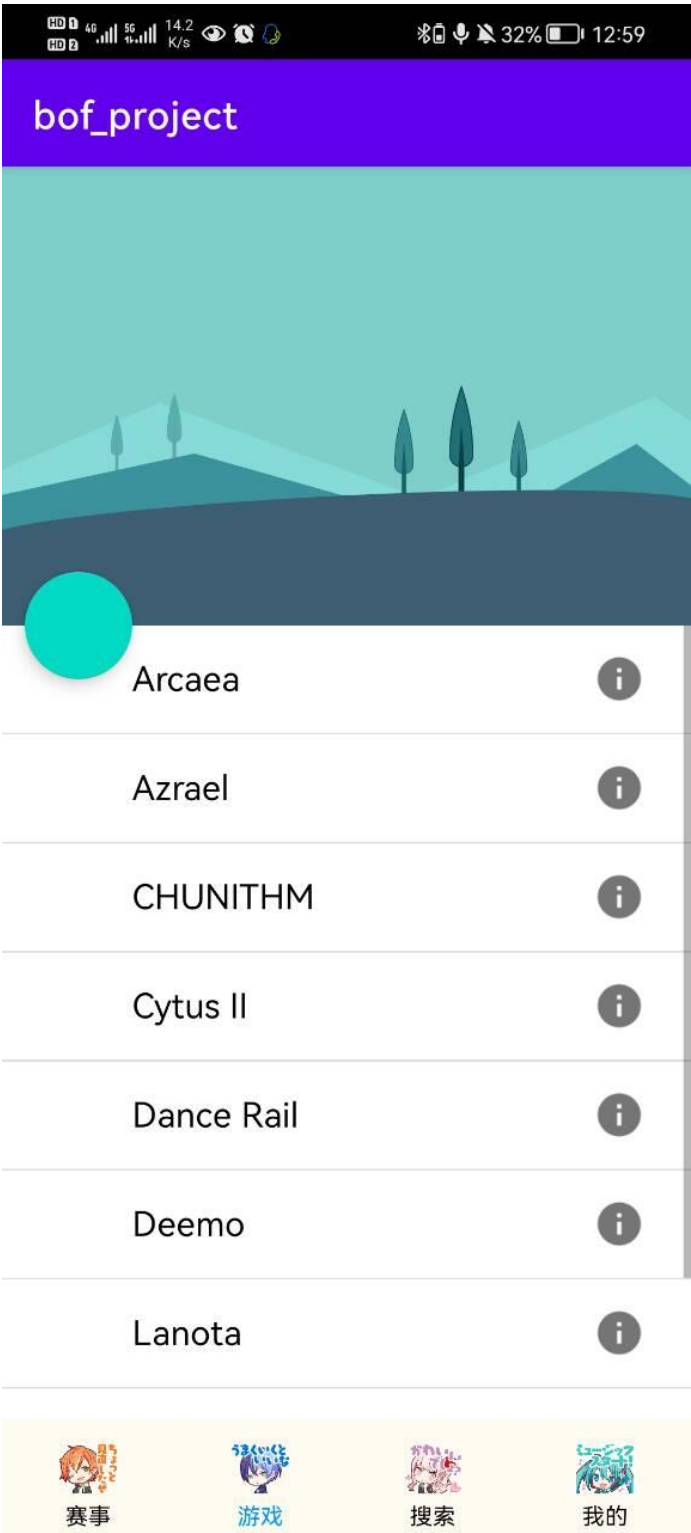
当前为参赛曲目界面。参赛曲目功能列举出该年参赛的所有作品，点击黄色的表项，该表项就可以展开。并可以点击“播放 BOF”查看每首作品的播放 BOF 的 bga 的通道，用户可以通过此通道来欣赏曲目的 BGA。

5.2.6 播放 BGA 界面实现



当前界面为 BGA 播放界面。该功能可以播放选定的 BOF 曲目，并可以选择是否收藏该曲目，便于在个人界面的收藏部分进行查看。点击左下角的“收藏”按钮，就可以收藏当前曲目，则按钮的文字会变成“取消收藏”。再次点击就会取消收藏，按钮恢复到原样。

5.2.7 游戏展示界面实现



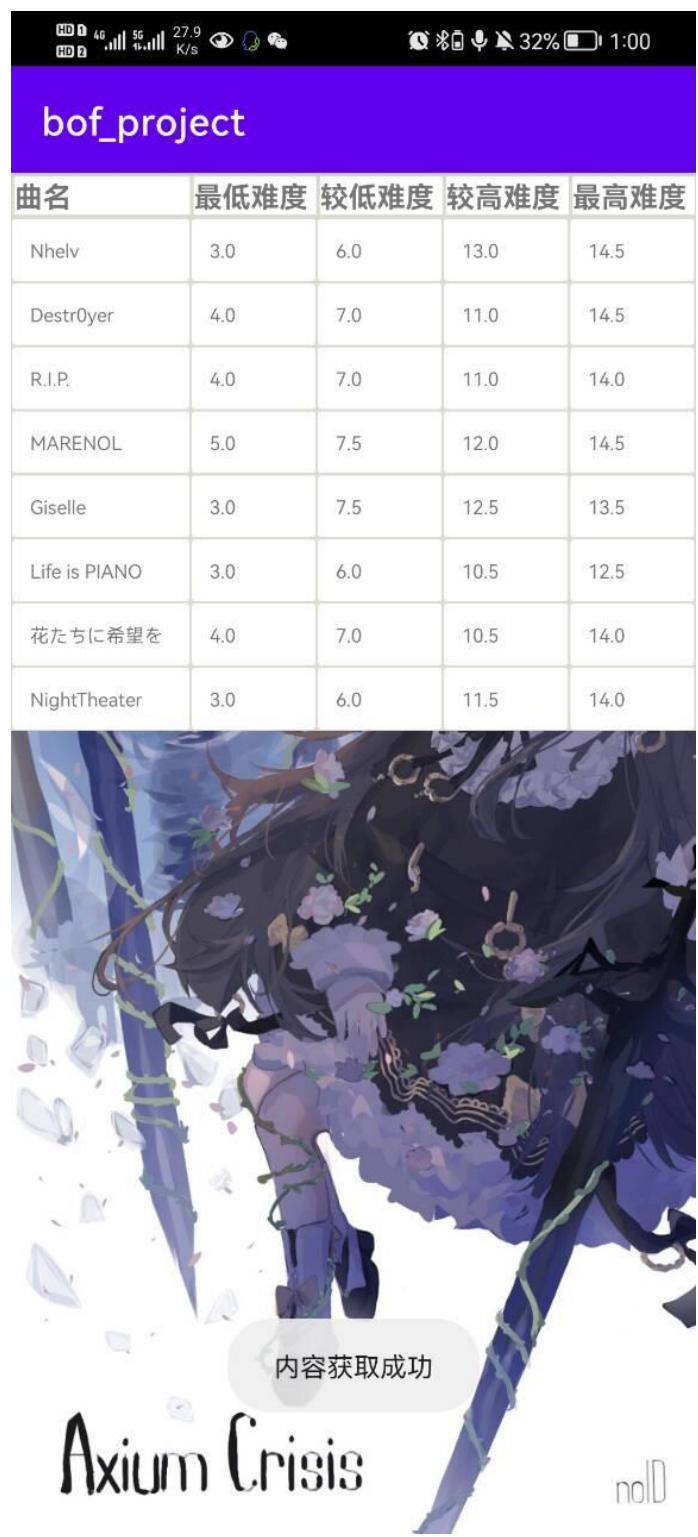
当前为游戏展示界面。该功能展示所有音乐游戏的列表，点击列表当中的游戏表项，可以跳转到游戏信息界面，查看更加详细的该游戏的信息。如果对该界面进行下拉操作，可以刷新列表，界面上方会出现刷新动画。

5.2.8 游戏信息界面实现



当前为游戏信息界面。该功能会展示某一个游戏的具体信息，包括游戏的名称、别名、发行日期、发行公司、支持平台和游戏简介等等。并提供查看收录曲目及其对应难度的表格的通道，点击“查看收录曲目及其对应难度”按钮即可。

5.2.9 游戏收录展示界面实现



当前为游戏收录展示界面。该功能会展示游戏中的曲目收录相关信息，包括游戏中曲目的各个难度阶级的具体难度，以表格的形式展现。

将音游中的各难度划分为最低难度、较低难度、较高难度和最高难度四种，收录了该曲目在当前音游中的难度标级。

5.2.10 搜索曲目界面实现



xi

搜索

Valhalla

xi

Aragami

xi

Dreadnought

Mastermind(xi+nora2r)

Maxi

Nizikawa

Bad Elixir

xi



赛事



游戏



搜索



我的

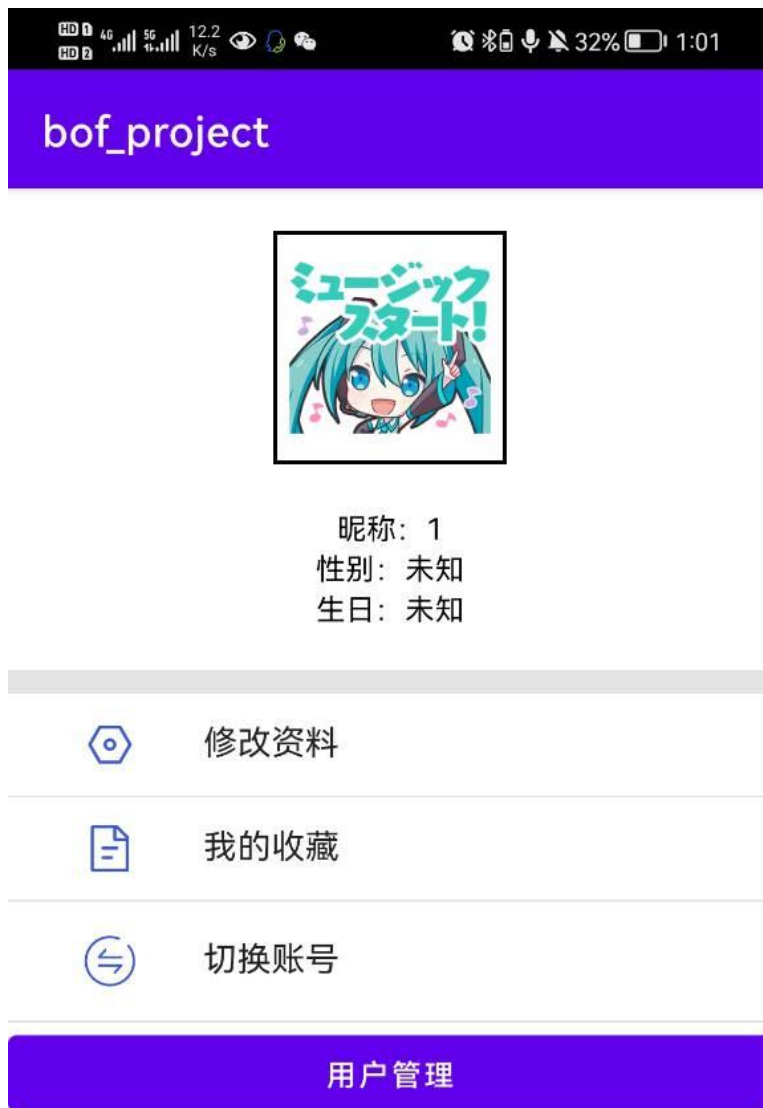
当前为搜索曲目界面。该功能可以输入关键词并搜索，对数据库中的曲目或者曲师进行匹配。对匹配到的搜索结果进行点击，就可以跳转到相应的播放界面，播放 BGA 视频。

5.2.11 个人中心界面实现



当前为个人中心界面。在界面上展示昵称、性别、生日等用户的信息，为普通用户提供修改资料、我的收藏、切换账号的功能，并为管理员提供用户管理的按钮，用于对用户账号进行操作。

点击“修改资料”表项，跳转到修改资料界面；点击“我的收藏”表项，跳转到我的收藏界面；点击“切换账号”表项，退出当前账号，回到登录界面。



登录成功

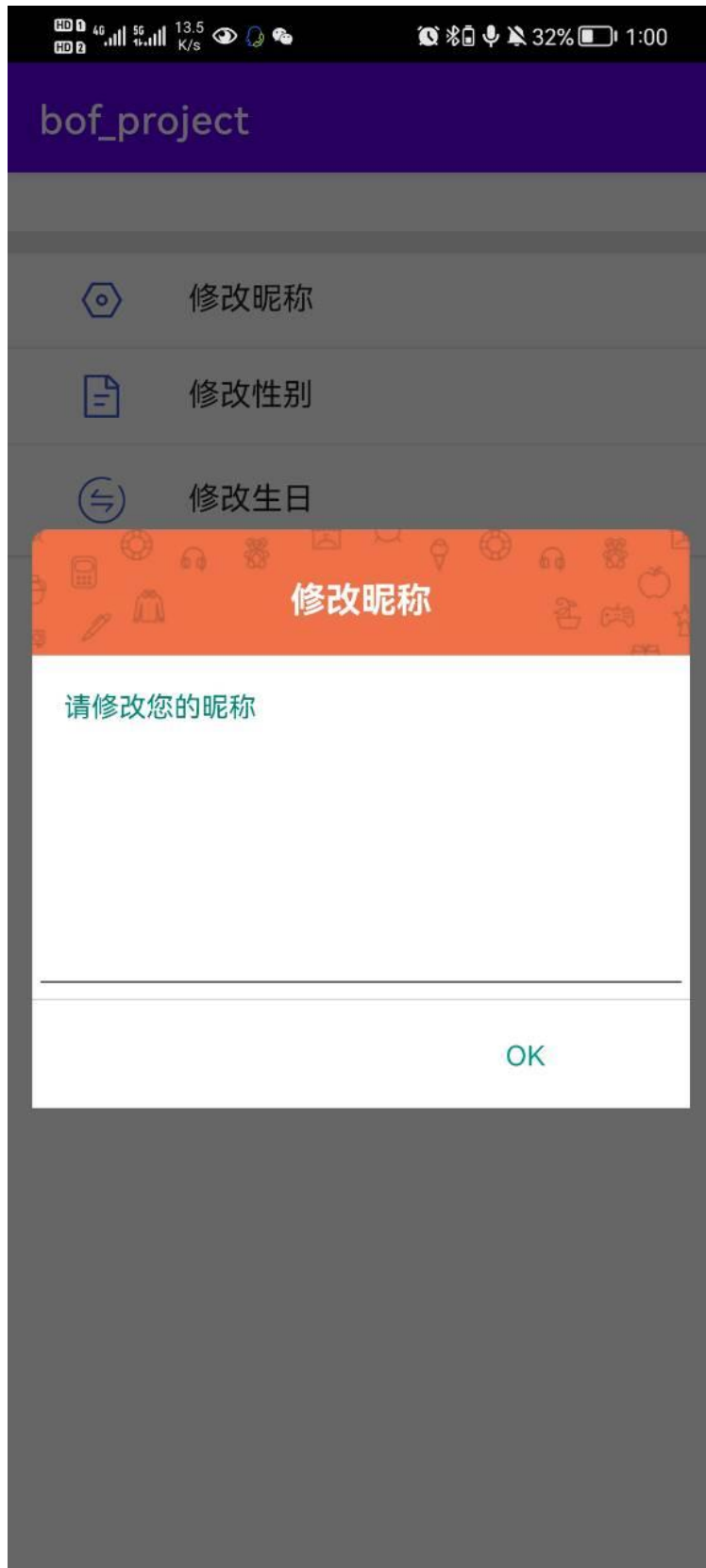


当处于管理员模式时，会显示“用户管理”按钮，点击以后进入用户管理界面，对用户进行操作。

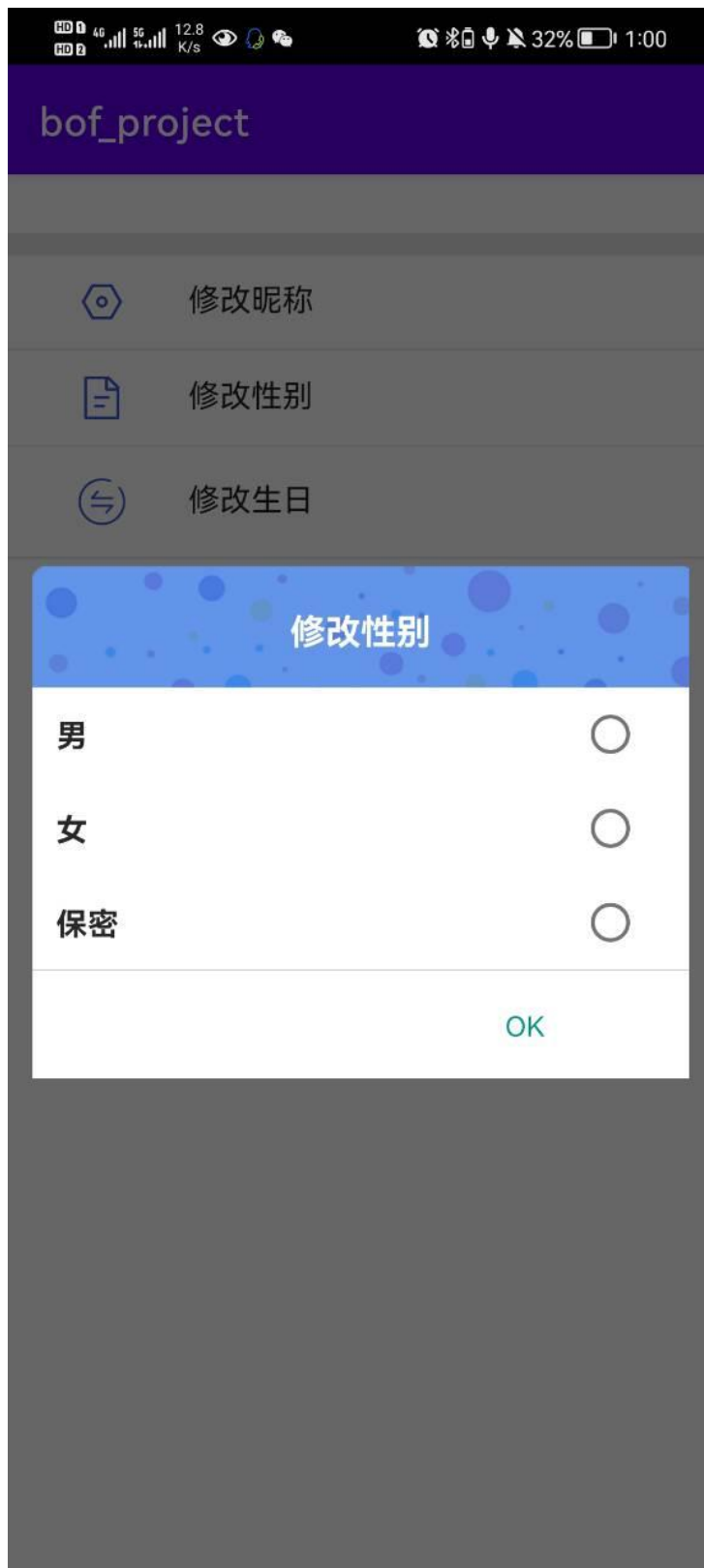
5.2.12 修改资料界面实现



当前为修改资料界面。该界面可以对用户的资料进行修改，包括昵称、性别和生日等。



点击“修改昵称”表项，会弹出对话框，提示输入修改的昵称。输入昵称后点击“OK”，就可以修改成功。

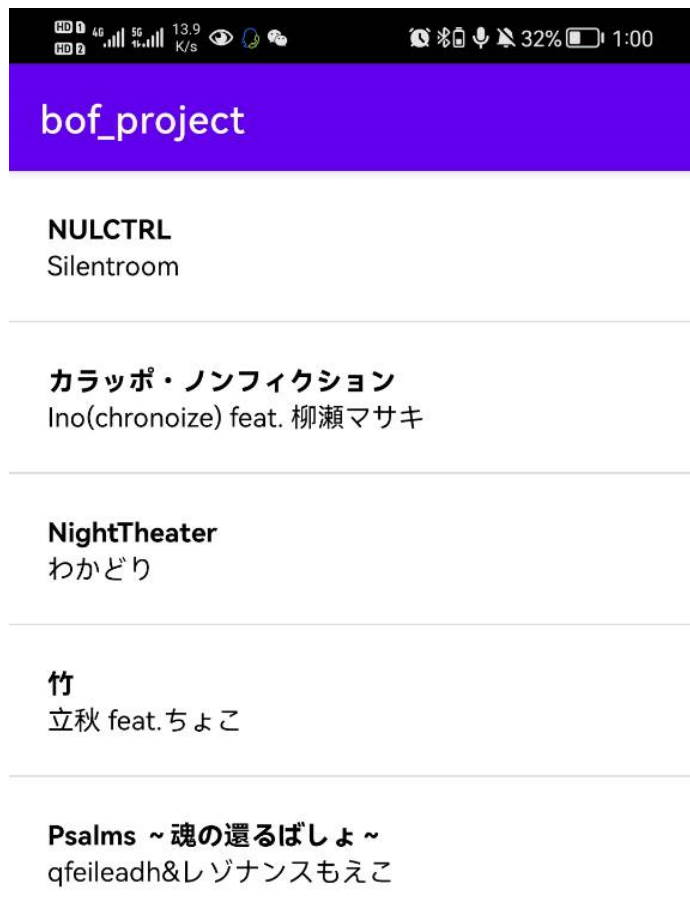


点击“修改性别”表项，会弹出对话框，提示修改性别。可以选择“男”、“女”和“保密”三种选项。选定以后点击“OK”就可以修改成功。



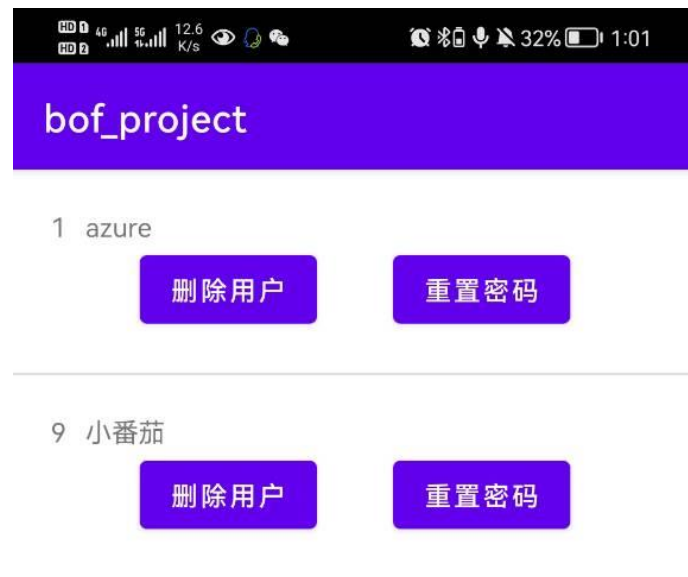
点击“修改生日”表项，会弹出对话框，提示修改生日。可以对弹出的日历进行日期选择，选择用户的生日。

5.2.13 我的收藏界面实现



当前为我的收藏界面。该界面可以查看用户收藏的曲目。用户点击某一个曲目，就可以播放该曲目的 BGA。

5.2.14 用户管理界面实现



用户管理功能仅限于管理员模式，普通用户无法使用。

该功能可以对普通用户进行直接管理，即删除用户和重置密码的功能，点击“删除用户”可以删除当前用户，点击“重置密码”就可以将密码改为初始密码。

6.3 后端系统实现

后端在 eclipse 上搭建 JDBC 的环境。

DBUtils.java: 进行有关数据库的操作

```
RegisterServ... web.xml DBUtils.java x InsertLikeSe... LoginServlet... JudgeLikeSer... DeleteUserSe... ResetPasswor... BaseBean.java 16
1 package database;
2
3 import java.sql.Connection;
4
5 public class DBUtils {
6     private Connection conn;
7     private String url = "jdbc:mysql://localhost/bof_database?characterEncoding=UTF-8&zeroDateTimeBehavior=convertToNull&allowMultiQueries=true";
8     private String user = "root"; // 用户名
9     private String password = "GWRAzure20010519"; // 用户密码
10    private Statement sta;
11    private ResultSet res; // 结果集
12
13    // 基础连接
14    public void openConnect()
15    {
16        try {
17            // 跨浏览器兼容性
18            Class.forName("com.mysql.cj.jdbc.Driver");
19            conn = DriverManager.getConnection(url, user, password); // 进行数据库连接
20            if (conn != null) {
21                System.out.println("连接成功!"); // 连接成功
22            }
23        } else {
24            System.out.println("数据库错误!");
25        }
26    } catch (ClassNotFoundException e) {
27        e.printStackTrace();
28        System.out.println("数据库错误!");
29    }
30 }
```

RegisterServlet.java: 进行具体的某个操作的 GET 和 POST。

```
RegisterServ... x web.xml SearchMusicS... DBUtils.java InsertLikeSe... LoginServlet... JudgeLikeSer... DeleteUserSe... ResetPasswor... 16
9 import javax.servlet.ServletException;
10 import javax.servlet.http.HttpServlet;
11 import javax.servlet.http.HttpServletRequest;
12 import javax.servlet.http.HttpServletResponse;
13 //import javax.xml.ws.Response;
14
15 import domain.BaseBean;
16 import domain.UserBean;
17 import database.DBUtils;
18 import com.google.gson.Gson;
19
20 public class RegisterServlet extends HttpServlet {
21
22    @Override
23    protected void doGet(HttpServletRequest request, HttpServletResponse response)
24        throws ServletException, IOException {
25        doPost(request, response);
26    }
27
28    @Override
29    protected void doPost(HttpServletRequest request, HttpServletResponse response)
30        throws ServletException, IOException {
31        {
32            System.out.println("request--->" + request.getRequestURL() + "====" + request.getParameterMap().toString());
33            String user_password = request.getParameter("user_password"); // 客户端传来注册密码
34            String user_nickname = request.getParameter("user_nickname"); // 客户端传来注册昵称
35            response.setContentType("text/html; charset=utf-8");
36            user_nickname = new String(user_nickname.getBytes("ISO8859-1"), "UTF-8");
37            if (user_password == "" || user_nickname == "")
```

▼ 16 servlet

- > 16 AlterBirthdayServlet.java
- > 16 AlterNicknameServlet.java
- > 16 AlterSexServlet.java
- > 16 DeleteUserServlet.java
- > 16 GetBofInfoServlet.java
- > 16 GetGameInfoServlet.java
- > 16 GetGameListServlet.java
- > 16 GetIncludeServlet.java
- > 16 GetLikeServlet.java
- > 16 GetMusicListServlet.java
- > 16 GetUserListServlet.java
- > 16 InsertLikeServlet.java
- > 16 JudgeLikeServlet.java
- > 16 LoginServlet.java
- > 16 RegisterServlet.java
- > 16 ResetPasswordServlet.java
- > 16 SearchMusicServlet.java

***Bean.java：服务端回传数据时使用的类。



另外，服务器端还有加载所需图片、视频等内容。

七、软件测试

7.1 数据库安全评估

尝试对 url 进行爆破。使用 Kali 上的 sqlmap。



尝试对库名的爆破，结果爆破失败，说明具备数据库安全。

```
root@kali: ~# sqlmap -u https://47.102.114.76/wp-login.php?id=2 --cookie wordpress_test_cookie=WP+Cookie=check --batch -db
[1.0.4EStable]
https://sqlmap.org

[!] Legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no
liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 23:41:45 /2022-06-14/

[23:41:45] [INFO] testing connection to the target URL
[23:41:46] [INFO] testing if the target URL content is stable
you provided a HTTP Cookie header value, while target URL provides its own cookies within HTTP Set-Cookie header which intersect with yours. Do you want to merge them in further requests? [Y/n] Y
[23:41:47] [INFO] target URL content is stable
[23:41:47] [INFO] testing if GET parameter 'id' is dynamic
[23:41:47] [WARNING] GET parameter 'id' does not appear to be dynamic
[23:41:48] [WARNING] heuristic (basic) test shows that GET parameter 'id' might not be injectable
[23:41:49] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[23:41:51] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[23:41:52] [INFO] testing 'MySQL > 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[23:41:54] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[23:41:57] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[23:41:59] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[23:42:00] [INFO] testing 'generic inline queries'
[23:42:03] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[23:42:05] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[23:42:07] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[23:42:09] [INFO] testing 'MySQL > 5.8.12 AND time-based blind (query SLEEP)'
[23:42:12] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[23:42:14] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[23:42:17] [INFO] testing 'Oracle AND time-based blind'
it is recommended to perform only basic UNION tests if there is not at least one other (potential) technique found. Do you want to reduce the number of requests? [Y/n] Y
[23:42:19] [INFO] testing 'generic UNION query (NULL) - 1 to 10 columns'
[23:42:25] [WARNING] GET parameter 'id' does not seem to be injectable
[23:42:27] [CRITICAL] all tested parameters do not appear to be injectable. try to increase values for '--level/' '--risk' options if you wish to perform more tests. if you suspect that there is some kind of prot
ection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent'

[*] ending @ 23:42:24 /2022-06-14/
```

7.2 端口检测

```
root@kali: ~# nmap -O 47.102.114.76
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-15 00:01 EDT
Nmap scan report for 47.102.114.76
Host is up (0.028s latency).
Not shown: 994 filtered tcp ports (no-response)
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
80/tcp    open  http
110/tcp   open  pop3
443/tcp   open  https
8080/tcp   open  sun-answerbook
Warning: OSscan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Actiontec M124WP-Q2531 WP (98%), DD-WRT V24-sp2 (Linux 5.4.37) (98%), Linux 4.4 (98%), Linux 3.2 (97%), Microsoft Windows XP SP3 or Windows 7 or Windows Server 2012 (95%), VMware Player
virtual NAT device (95%), BlueArc Titan 2100 NAS device (95%), Microsoft Windows XP SP3 (94%), Pirelli DP-10 VoIP phone (91%)
No exact OS matches for host (test conditions non-ideal).
OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 52.35 seconds
```

查看与设置开启的端口一致，说明端口安全，遵循最小开放原则。

7.3 漏洞检测

使用 nikto 生成漏洞报告。

```
root@kali: ~# nikto -h 47.102.114.76 -o vulnerabilities.html
- Nikto v2.1.6

+ Target IP: 47.102.114.76
+ Target Hostname: 47.102.114.76
+ Target Port: 80
+ Start Time: 2022-06-15 00:08:14 (GMT-4)

+ Server: Apache
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ Uncommon header 'x-redirect-by' found, with contents: WordPress
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Root page / redirects to: https://huahuadan.xyz/
+ Uncommon header 'link' found, with contents: <https://huahuadan.xyz/wp-json/>; rel="https://api.w.org/"
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Entry '/wp-admin/' in robots.txt returned a non-forbidden or redirect HTTP code (302)
+ 'robots.txt' contains 2 entries which should be manually viewed.
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ OSVDB-62684: /wp-content/plugins/hello.php: The WordPress hello.php plugin reveals a file system path
+ /wp-links-opml.php: This WordPress script reveals the installed version.
+ OSVDB-3092: /license.txt: License file found may identify site software.
+ /wp-app.log: Wordpress' wp-app.log may leak application/system details.
+ /wordpresswp-app.log: Wordpress' wp-app.log may leak application/system details.
+ 7920 requests: 0 error(s) and 13 item(s) reported on remote host
+ End Time: 2022-06-15 01:05:37 (GMT-4) (3443 seconds)
```

打开 kali 目录下的 vulnerabilities.html，就可以查看目标靶机的漏洞报告。

47.102.114.76 / 47.102.114.76 port 80	
Target IP	47.102.114.76
Target hostname	47.102.114.76
Target Port	80
HTTP Server	Apache
Site Link (Name)	http://47.102.114.76.80/
Site Link (IP)	http://47.102.114.76.80/
URI	/
HTTP Method	GET
Description	The anti-clickjacking X-Frame-Options header is not present.
Test Links	http://47.102.114.76.80/ http://47.102.114.76.80/
OSVDB Entries	OSVDB-0
URI	/
HTTP Method	GET
Description	The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
Test Links	http://47.102.114.76.80/ http://47.102.114.76.80/
OSVDB Entries	OSVDB-0
URI	/
HTTP Method	GET
Description	Uncommon header 'x-redirect-by' found, with contents: WordPress
Test Links	http://47.102.114.76.80/ http://47.102.114.76.80/
OSVDB Entries	OSVDB-0
URI	/
HTTP Method	GET
Description	The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
Test Links	http://47.102.114.76.80/ http://47.102.114.76.80/
OSVDB Entries	OSVDB-0

八、系统运行维护

在服务器端，系统的 Tomcat 服务器始终在 Eclipse 平台上运行，因此随时可以使用安卓客户端，与服务端端的数据库进行交互。

在 Tomcat 服务器的运行过程中，Servlet 之间相互独立，因此在编写或者修改一个 Servlet 的时候，并不影响其他功能的 Servlet 的使用，运行的代价较小。

Tomcat 服务器上随时可以在资源文件夹中加入图片或者视频资源，利于维护和更新。