

简易的动画点评网站(番组X)

组员	学号	邮件
陈梓立	1500012726	wander4096@gmail.com
叶志晟	1500012804	yzs981130@126.com

引言

本课程项目实现了一个简易的动画点评网站(bangumix)，在网站上可以浏览动画的本身的基本信息，看到其他用户为动画贴的标签和给动画打的评分以及评论，可以注册用户并参与到添加新的动画信息，给已有动画评分，添加标签和评论中。

选择这个项目的�原因主要是出于小组成员对动画领域的热爱。最初的选择是实现一个个人博客系统，但感觉到博客系统对应的数据库模式较为枯燥，且自己不会使用自己开发的博客系统；然而这样一个动画点评网站虽然简单，却有足够的热情去实现自己真正想要的点评功能。

主要工作包括设计表示网站信息的数据库模式，实现应用程序与数据库的交互逻辑，实现前后端的交互逻辑。

项目网站的前端部分在叶志晟同学的设计和实现下非常优雅美观；应用程序部分在陈梓立同学的设计和实现下避免了传统 J2EE 和 Spring 框架设计时过于臃肿和过度抽象的问题，代码分层简洁，内容清楚；数据库模式在陈梓立同学的设计和实现下实践了若干项高级数据库特性(触发器，存储过程，视图等)。

完成这个项目，小组成员对如何开发一个完整的可独立运行的网站有了整体的了解，对应用程序和数据库的交互有了丰富的实践，同时也体会到了数据库高级功能的表达能力和数据库设计中表模式设计的重要性。

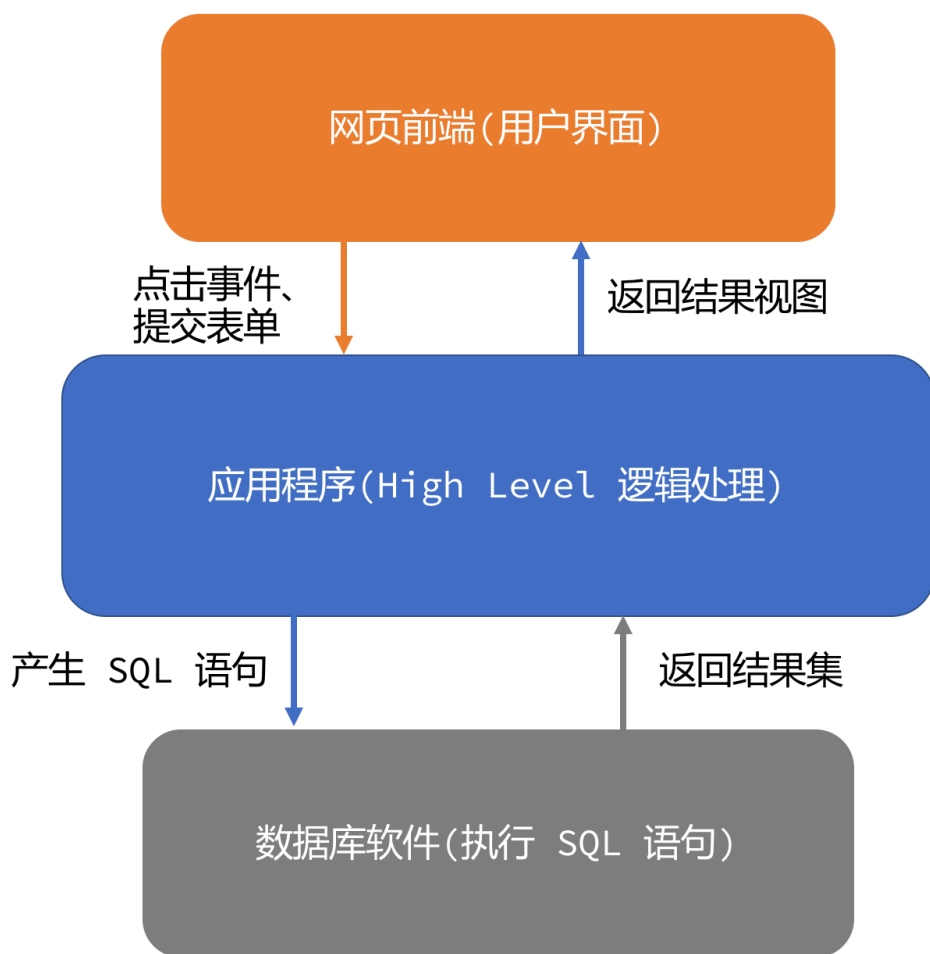
相关工作

项目采用 J2EE 的技术搭建网站。

1. MySQL 作为后端数据库软件
2. Kotlin 作为后端开发语言
3. Spring Boot 作为后端开发框架
 1. Spring MVC 作为整体前后端交互结构
 2. Spring JdbcTemplate 作为与数据库交互的手段
4. Thymeleaf 作为页面渲染框架
5. Semantic UI 作为前端 UI 设计框架

系统框架

宏观视角

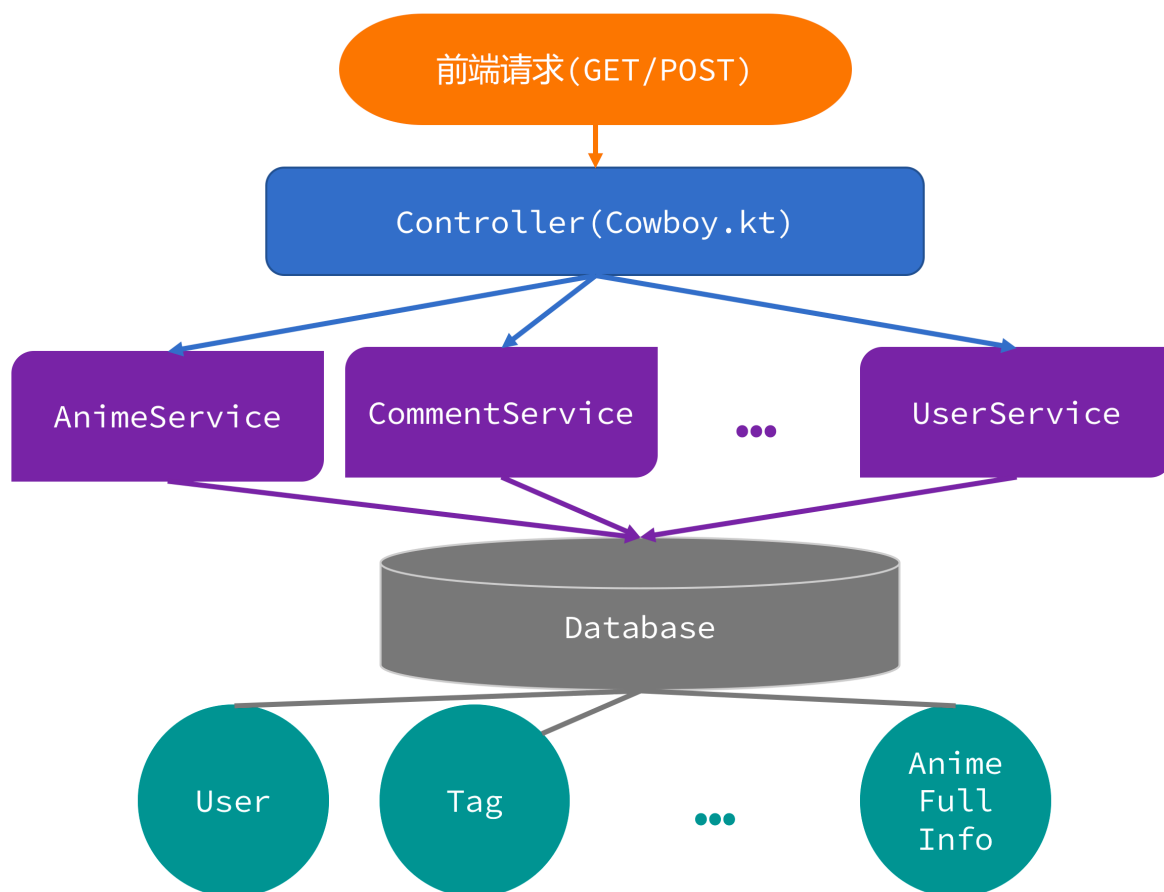


整体来看，B/S 服务包括三个组成部分：

1. 网页前端作为用户界面，将后台产生的结果视图以友好的方式给出，用户在这一层通过点击事件或提交表单发送自己的请求
2. 应用程序作为后端的一部分，负责处理前端发送过来的事件，经过验证，转化和分解等一系列步骤后，根据需求生成相应的 SQL 语句，向数据库提出查询或进行更新，获得所有需要的信息后生成结果视图并返回到前端显示
3. 数据库软件，本项目中使用 MySQL，内部存有事先设计的数据库模式(包括表，视图，触发器和存储过程等)和样例数据，在应用程序通过数据库连接器传来 SQL 请求后执行相应的 SQL 并对查询语句返回相应的结果集

应用程序结构

三个组成部分中，前端页面使用 thymeleaf 渲染模板渲染生成，与应用程序部分结合紧密，数据库的详情在数据库模式一节介绍，因此这里对应用程序部分做进一步的拆分

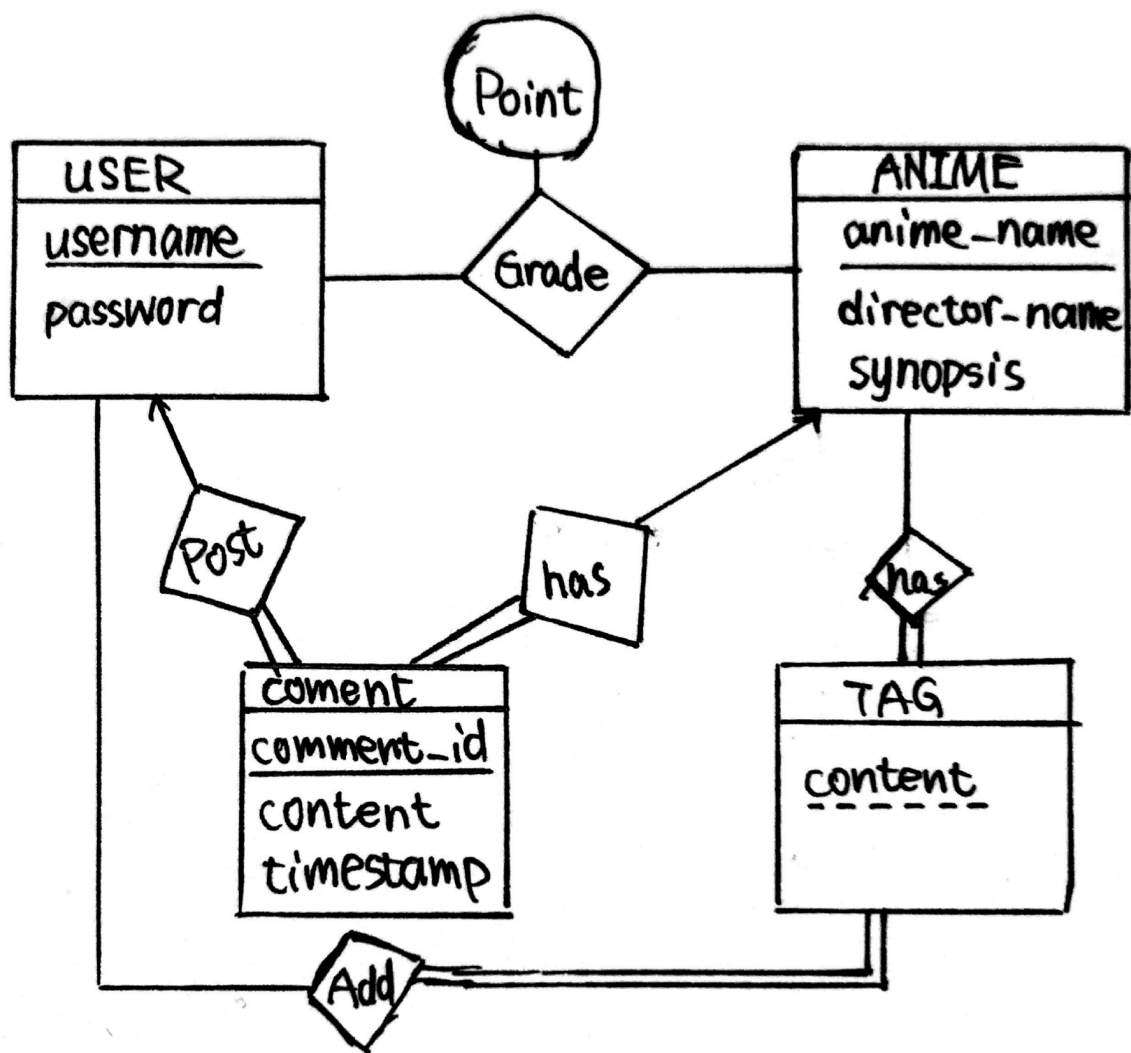


设计上，模仿了传统的 MVC 的结构。

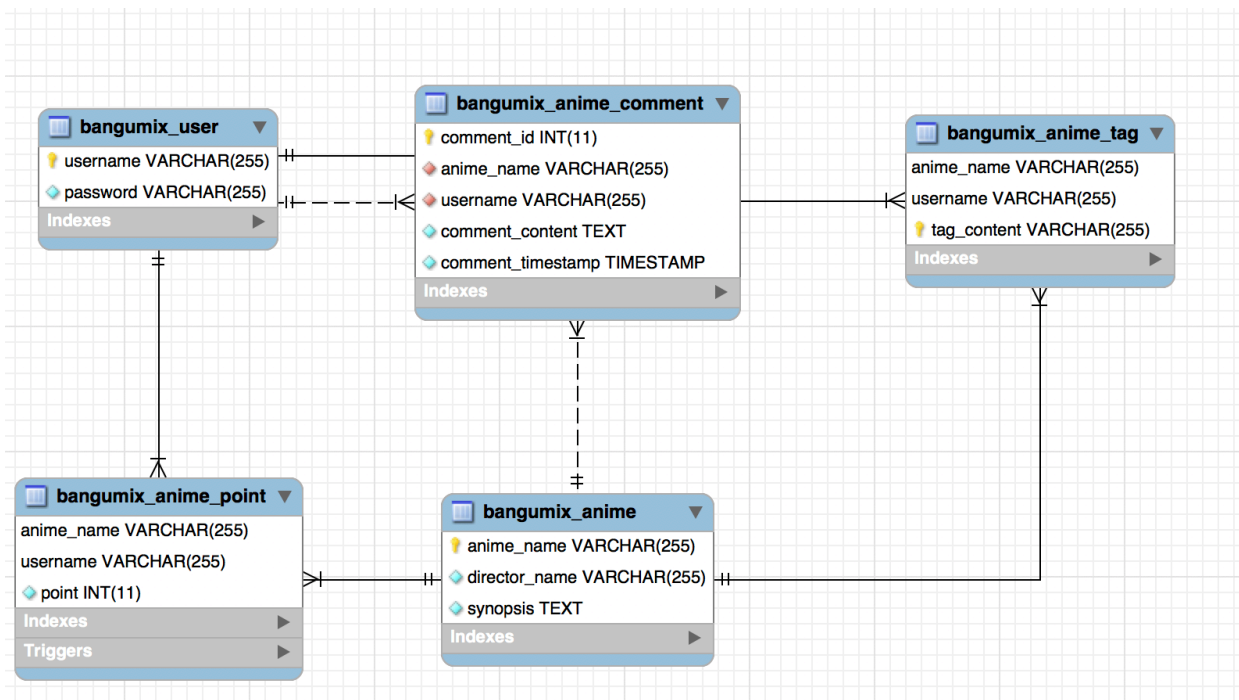
1. 对于前端的请求，经过 Controller 层分发到具体的处理函数中。不同于传统的 Controller 层多个 Controller 处理不同请求，考虑到本项目总体需要处理的请求种类不多，将 Controller 的功能压缩到单个类中 (Cowboy.kt)。
2. Cowboy 类中的各个函数分别处理不同的前端请求，通过与 Service 层交互完成需要数据库系统参与的逻辑，拿到所有需要产生结果视图的数据后生成结果视图并返回前端
3. 不同于传统的 DAO 层，将具体的数据库服务内容上升到 Service 层，将数据库返回的结果集转化为 POJO 的内容下放到 Model 层
4. 从数据库的具体表中和从视图中取得的数据，其 model 都放在 Model 层中，故也可叫 POJO 层，避免了大型 J2EE 程序中过分区分 Entity, VO, DTO, BO 的负担

数据库模式

E-R 图



模式描述



- bangumix_user 表

主键为 username。没有外键约束，显然是 BCNF 的

- bangumix_anime 表

主键为 anime_name。没有外键约束，director_name 和 synopsis 之间没有关系，是 BCNF 的

- bangumix_anime_point 表

主键为 (anime_name, username)，分别为前两个表对应属性的外键，anime_name 和 username 之间没有关系，point 由两者共同决定，是 BCNF 的

- bangumix_anime_comment 表

主键为 comment_id。anime_name 和 username 如前是对应表中属性的外键，考虑到 comment_timestamp 在 1 毫秒之内无法区分，表是 BCNF 的；如果假定 timestamp 可以准确区分，则存在 comment_content 对码的传递依赖，从而是 2NF 的

- bangumix_anime_tag 表

主键为 (anime_name, username, tag_content)。anime_name 和 username 如前是对应表中属性的外键，同样的 username 和 anime_name 可以有多个 tag_content，且 username 和 anime_name 没有依赖，因此是 BCNF 的

- 视图

动画 (anime) 的完整信息包括动画名称，动画导演，动画概要，已有标签和平均评分，在 bangumix_anime 表中只有前三项信息，因此设计了视图 bangumix_anime_full_info 来观察完整的动画信息。同时，为了支持排序显示和推荐动画的服务，将 full info 的过程进行具体拆分，一共编写了三个视图

```
create view bangumix_anime_rank as
  select anime_name, avg(point) as rank
  from bangumix_anime_point
  group by anime_name
  order by rank desc
;

create view bangumix_anime_tag_count as
  select anime_name, tag_content, count(tag_content) as times
  from bangumix_anime_tag
  group by anime_name, tag_content
  order by times desc
;

create view bangumix_anime_full_info as
  select x.anime_name, x.director_name, x.synopsis, x.rank,
  group_concat(tag_content separator '/') as tags
  from (
    select a.anime_name, a.director_name, a.synopsis, b.rank
    from (bangumix_anime as a
```

```

        left join bangumix_anime_rank as b on a.anime_name =
b.anime_name)) as x
        left join bangumix_anime_tag_count as t on x.anime_name =
t.anime_name
        group by x.anime_name, x.director_name, x.synopsis, x.rank
        order by x.rank desc
;

```

- 触发器

MySQL 不支持表的 check 检查，具体来说，可以编写 check 检查，但在插入过程中不会有实际检查的动作，考虑使用触发器来检查插入的数据。实际上，数据已在应用程序层检查过，且一般的非空要求可以通过 `not null` 来约束。这里针对评分的要求 check 分数是否为 1 到 5 之间的整数

```

drop trigger if exists insert_bangumix_anime_point;
delimiter //
create trigger insert_bangumix_anime_point
before
insert on bangumix_anime_point
for each row begin
    if new.point > 5 or new.point < 1
    then
        CALL `Cannot add or update row: point such be an integer in
[1, 5]`;
    end if;
end; //
delimiter ;

```

- 存储过程

我们把为用户推荐动画的过程写成存储过程，具体逻辑是获得用户打过的标签，提取相同的标签的动画，从中选取前 3 个，不足 3 个用所有动画的表补全

```

drop procedure if exists bangumix_recommend;
delimiter //
create procedure bangumix_recommend(
    in_username varchar (255)
)
begin
    select anime_name
    from (select s.anime_name, min(s.tag) as prec
        from (select anime_name, tag
            from (select distinct username, anime_name, 0 as tag
                from
                    (bangumix_anime_tag as a)
                natural join
                    (select distinct tag_content

```

```

        from bangumix_anime_tag as b
        where b.username = in_username) as x) as y
    where not exists (select *
                      from bangumix_anime_tag as z
                      where z.username = in_username and
z.anime_name = y.anime_name)

    union

    (select distinct anime_name, 1 as tag
     from bangumix_anime)) as s
group by s.anime_name
order by prec
limit 3) as t;

end; //
delimiter ;

```

操作实现

1. 首页上显示所有已有动画的完整信息，通过 `select * from bangumix_anime_full_info` 实现；首页另外在用户登录时使用上面的存储过程给出为用户推荐的动画列表

```

// Cowboy.kt
@GetMapping("/")
model.addAttribute("animes", animeService.findAllFullInformation())

// AnimeService.kt
fun findAllFullInformation(): List<AnimeFullInfo> {
    return jdbcTemplate.query("select * from bangumix_anime_full_info",
AnimeFullInfo())
}

```

为你推荐

[CODE GEASS 反叛的鲁路修](#) | [Fate/stay night \[Unlimited Blade Works\]](#)
[JOJO的奇妙冒险: 不灭钻石](#)

动画名	导演	概要	分数	标签
CODE GEASS 反 叛的鲁路 修	谷 口 悟 朗	作品描述被神圣不列颠帝国侵略而亡国的日本——11区中，在生死边缘获得神秘魔女c.c.赋予的Geass力量，立志要粉碎帝国的黑色王子鲁路修·兰佩路基，与坚持公理的白色骑士枢木朱雀所掀起的巨大变化。	5.0	
Fate/kaleid liner 魔法 少女☆伊 莉雅	大 沼 心	故事主要讲述了女主角伊莉雅是就读穗群原学园的普通女生，在某天遇到了自称人工天然精灵的魔法露比万花筒之杖，并强制地被缔结契约，成为了魔法少女伊莉雅。而已是红宝石之星持有人的她，还变成了万花筒之杖原持有者的	5.0	

2. 用户可以添加新的动画信息

```
// Cowboy.kt
@PostMapping("/add/anime")
animeService.insertOne(Anime(animeName, directorName, synopsis))

// AnimeService.kt
fun insertOne(anime: Anime) {
    when {
        !validateAnime(anime) -> throw IllegalArgumentException("动画信息  
格式有误, 各项不可为空! ")
        null != findByName(anime.animeName) -> throw  
IllegalArgumentException("同名动画已存在, 新增失败! ")
        else -> jdbcTemplate.update("insert into bangumix_anime  
(anime_name, director_name, synopsis) values (?, ?, ?)", anime.animeName,  
anime.directorName, anime.synopsis)
    }
}
```


动画名:

DARING in the FRANXX

导演:

锦织敦史

概要:

我要和你在一起!

提交

3. 用户可以添加和删除标签

```
// Cowboy.kt
@PostMapping("/add/tag")
tagService.insertOne(animeName, session.getAttribute("user") as String,
animeTag)

@PostMapping("/remove/tag")
@ResponseBody
tagService.removeOne(animeName, session.getAttribute("user") as String,
animeTag)

// TagService.kt
fun insertOne(animeName: String, username: String, tagContent: String)
{
    validate(animeName, username, tagContent)
    jdbcTemplate.update("insert into bangumix_anime_tag (anime_name,
username, tag_content) values (?,?,?)", animeName, username, tagContent)
}

fun removeOne(animeName: String, username: String, tagContent: String)
{
    jdbcTemplate.update("delete from bangumix_anime_tag where
anime_name=? and username=? and tag_content=?", animeName, username,
tagContent)
}
```

Fate/kaleid liner 魔法少女☆伊莉雅

导演: 大沼心

概要: 故事主要讲述了女主角伊莉雅是就读穗群原学园的普通女生，在某天遇到了自称人工天然精灵的魔法露比万花筒之杖，并强制地被缔结契约，成为了魔法少女伊莉雅。而已是红宝石之星持有人的她，还变成了万花筒之杖原持有者的魔术师远坂凛的奴隶，在她的命令之下，被迫帮忙回收沉睡于冬木市的某危险的卡片.....

系统标签

萌！无敌！ Fate

你的标签

Fate 萌！无敌！ Enter tags 添加标签

系统评分

♥♥♥♥♥

4. 用户可以添加评论

```
// Cowboy.kt
@PostMapping("/add/comment")
commentService.insertOne(AnimeComment(0, animeName,
session.getAttribute("user") as String, commentContent))

// CommentService.kt
fun insertOne(animeComment: AnimeComment) {
    if (validateAnime(animeComment)) {
        jdbcTemplate.update("insert into bangumix_anime_comment
(anime_name, username, comment_content, comment_timestamp) values
(?,?,?,?)", animeComment.animeName, animeComment.username,
animeComment.commentContent, Timestamp(System.currentTimeMillis()))
    } else {
        throw IllegalArgumentException("评论内容不可为空!")
    }
}
```

用户	评论	评论时间
Tison	美游真的是太萌啦 (2018-05-26 20:16:02.0
Real	伊莉雅伊莉雅upup	2018-05-26 20:16:02.0
Real	美游也很努力啦	2018-05-26 20:16:02.0

评论:

提交

5. 用户可以给动画打分

```
// Cowboy.kt
@PostMapping("/add/point")
rankService.insertOne(animeName, session.getAttribute("user") as
String, point)

// RankService.kt
fun insertOne(animeName: String, username: String, pointAsString:
String) {
    val point = validatePoint(pointAsString)
    if (alreadyPoint(animeName, username)) { // Update
        jdbcTemplate.update("update bangumix_anime_point set point = ?
where anime_name = ? and username = ?", point, animeName, username)
    } else { // Insert
        jdbcTemplate.update("insert into bangumix_anime_point
(anime_name, username, point) values (?, ?, ?)", animeName, username, point)
    }
}
```

系统评分



你的评分: 5

分数



修改评分


6. 用户可以登录


```
// Cowboy.kt
@PostMapping("login")
if (userService.checkUsername(username) != null) {
    if (userService.checkUsernameAndPassword(username, password.md5()) != null) {
        session.setAttribute("user", username)
        return "redirect:/"
    } else {
        redirect.addFlashAttribute("errorMessage", "密码错误! ")
        return "redirect:/login"
    }
} else {
    redirect.addFlashAttribute("errorMessage", "用户不存在! ")
    return "redirect:/login"
}

// UserService.kt
fun checkUsername(username: String): User? {
    val maybeUser = jdbcTemplate.query("select * from bangumix_user
where username = ?", arrayOf(username), User())
    return maybeUser.firstOrNull()
}

fun checkUsernameAndPassword(username: String, password: String): User?
{
    val maybeUser = jdbcTemplate.query("select * from bangumix_user
where username = ? and password = ?", arrayOf(username, password), User())
    return maybeUser.firstOrNull()
}
```

登录

 Tison

 ●●●●●●

Login


还没有账号? [点击注册](#)

7. 可以注册新用户


```
// Cowboy.kt
@PostMapping("/register")
if (password != repeatPassword) {
    redirect.addFlashAttribute("errorMessage", "两次输入的密码不一致! ")
    return "redirect:/register"
} else {
    try {
        userService.registerUser(User(username, password.md5()))
    } catch (e: IllegalArgumentException) {
        redirect.addFlashAttribute("errorMessage", e.message)
        return "redirect:/register"
    }
}

// UserService.kt
fun registerUser(user: User) {
    if (null != checkUsername(user.username)) {
        throw IllegalArgumentException("用户已存在")
    } else {
        jdbcTemplate.update("insert bangumix_user (username, password)
values (?, ?)", user.username, user.password)
    }
}
```

注册

 pku_o9kg





Register

已有账号? [点击登录](#)

总结

本项目是基于小组成员的爱好兴趣实现的，完成了基本的 B/S 服务的完整内容。前端非常美观，应用程序在结构设计上相对于传统 MVC 模式有自己的思考和改进，数据库相关内容实践了多个高级特性。总体项目实现了实习的牢固数据库概念基本知识，增强安装，操作数据库的能力，提高对数据库的感性认识的目标。