课 程 论 文

|  |  |
| --- | --- |
| **课程名称：** | **现代软件开发技术** |
| **学期：** | **2021-2022学年第2学期** |
| **项目名称：** | **基于图片分享的社交平台** |
| **年级专业：** | **2019级软件工程专业** |
| **学号：** |  |
| **姓名：** |  |
| **学号：** |  |
| **姓名：** |  |
| **授课教师：** | **肖磊** |
| **提交日期：** | **2022年05月25日** |
| **论文成绩：** |  |
| **项目成绩：** |  |
| **总成绩：** |  |

华南农业大学 数学与信息学院

**华南农业大学数学与信息学院**

**《现代软件开发技术》 软件项目评分表**

|  |  |
| --- | --- |
| **项目名称** | 基于图片分享的社交平台 |
| **团队成员** | 学号，姓名 ； 学号，姓名 |
| **开发技术** | 简要说明使用的开发技术 |
| |  |  |  |  | | --- | --- | --- | --- | | 规定功能 | 完成情况 | 选做及增加功能 | 完成情况 | | 01显示分享列表 |  | 01限制词管理 |  | | 02搜索分享 |  | 02限制词检测 |  | | 03点赞榜 |  | 关注用户 |  | | 04最新发布 |  | 我的关注 |  | | 05查看分享 |  | 我的粉丝 |  | | 06登录/注册 |  | 用户设置 |  | | 07点赞分享 |  | 搜索用户 |  | | 08收藏分享 |  | 查看用户主页 |  | | 09评论分享 |  | 热搜榜 |  | | 10回复评论 |  | 密码加密 |  | | 11我的分享 |  |  |  | | 12发布分享 |  |  |  | | 13修改分享 |  |  |  | | 14分享状态 |  |  |  | | 15我的收藏 |  |  |  | | 16注册用户管理 |  |  |  | | |
| **评语：** | |
| **项目成绩** |  |

目录

**[1. 问题描述与需求分析 1](#_Toc15345)**

[1.1. 项目基本情况 1](#_Toc7943)

[1.2. 功能说明 1](#_Toc11889)

[1.3. 软件技术、开发工具和开发环境 4](#_Toc27678)

**[2. 系统设计 5](#_Toc28041)**

[2.1. 总体设计方案 5](#_Toc32610)

[2.2. 数据库设计 6](#_Toc30527)

[2.3. 界面设计 9](#_Toc30817)

[2.4. 软件系统结构设计 9](#_Toc10236)

**[3. 系统实现 11](#_Toc30865)**

[3.1. 系统项目结构 11](#_Toc16208)

[3.2. 系统全局配置说明 12](#_Toc32137)

[3.3. 层次说明 12](#_Toc6061)

**[4. 软件测试 21](#_Toc26177)**

**[5. 项目总结 21](#_Toc4407)**

[5.1. 项目实现的功能 21](#_Toc17202)

[5.2. 改进空间 21](#_Toc28635)

[5.3. 个人心得体会 21](#_Toc20614)

#### 问题描述与需求分析

##### 项目基本情况

##### 功能说明

###### 显示分享列表

以分页的形式显示分享列表。列表中的每个分享要展示分享的内容、图片，发布分享用户的用户名、头像，分享发布的时间，分享的点赞数、评论数和收藏数，以及当前用户是否已点赞收藏该分享。点击分享内容跳转到分享详情页面，点击分享图片展示大图，点击用户头像可跳转到用户主页。根据用户是否已点赞、收藏该分享，点赞按钮、收藏按钮以不同颜色显示。点击评论按钮可展示用户评论列表，再次点击可将用户评论列表折叠。点击评论中的查看回复按钮可查看该评论下的回复。

###### 搜索分享

在搜索框中输入内容，点击搜索按钮或按下回车键，以字符串模糊匹配的方式对分享的内容进行搜索，并将匹配的分享以分享列表的形式分页展示。

###### 点赞榜

按照点赞数降序展示分享列表。

###### 最新发布

按照发布时间最新的顺序展示分享列表。

###### 查看分享

点击分享列表中分享的内容进入分享详细页面。

###### 登录/注册

点击注册按钮进入注册页面，用户输入用户名、密码、确认密码，若用户名重复或确认密码与密码不匹配，则不允许注册，若合法，则点击注册按钮进行注册。点击登录按钮进入登录页面，输入用户名和密码进行登录，用户名存在且密码正确则登录成功。

###### 点赞分享

注册用户可点赞分享。分享的点赞按钮根据用户是否已点赞以不同的颜色展示。若未点赞，点击点赞按钮对分享点赞，若已点赞，点击点赞按钮对分享取消点赞。

###### 收藏分享

注册用户可收藏分享。分享的收藏按钮根据用户是否已收藏以不同的颜色展示。若未收藏，点击收藏按钮收藏分享，若已收藏，点击收藏按钮对分享取消收藏。

###### 评论分享

注册用户可评论分享。用户点击分享的评论按钮后，展示评论输入框，用户在输入框中输入评论内容，点击发表按钮发表评论。新发表的评论实时展示在评论列表中。每条评论需要展示出评论的内容，发布评论的用户的用户名、头像，评论发布时间以及回复数。

###### 回复评论

注册用户可回复评论。用户点击评论中的回复按钮后，展示回复输入框，用户在输入框中输入回复内容，点击提交按钮提交回复。实时更新评论的回复状态。每条回复需要展示出回复的内容，提交回复的用户的用户名、头像，回复提交时间。

注册用户可对回复进行回复。用户点击回复中的回复按钮进行回复。与对评论进行回复不同的是，对回复的回复在回复的内容前会显示“回复@xxx”，以明确所回复的用户。

###### 我的分享

以分页分享列表的形式展示当前用户发布的分享。包含分享内容、用户名、用户头像、发布时间信息。点击详情进入分享详情页面，点击编辑按钮展示分享编辑页面、点击删除按钮删除已发布的分享。

###### 发布分享

注册用户可发布分享。进入分享发布页面，可输入分享内容、设置分享是否公开、以及上传图片，已选择的图片将会展示出来，每个分享最多可有6个图片。点击发布按钮发布分享。

###### 修改分享

在分享修改页面中，用户可对分享的内容、分享是否公开、以及分享的图片进行编辑，点击提交按钮确认修改。

###### 分享状态

分享的状态分为“仅自己可见”和“公开”，在发布分享时可进行设置，在分享发布后，可在修改分享页面中进行修改。

###### 我的收藏

以分页分享列表的形式展示当前用户收藏的分享。包含分享内容、用户名、用户头像、发布时间信息。点击详情进入分享详情页面，点击删除按钮可取消收藏。

###### 注册用户管理

管理员可以禁⽌或重新允许⼀个注册⽤户进⾏分享活动。处于禁⽌状态的注册⽤户的分享均不可⻅。

###### 限制词管理

管理员可对限制词进行管理，可添加或删除限制词。

###### 限制词检测

在用户发布分享或修改分享内容前，对分享的内容进行限制词检测，若分享内容中存在限制词，则不允许用户发布或修改分享内容。

###### 关注用户

注册用户可关注用户。进入其他用户的主页，根据用户是否已关注，关注按钮显示为“关注”或“取消关注”，点击按钮可关注用户或取消关注用户。

###### 我的关注

注册用户可查看我的关注列表，列表中展示当前用户已关注用户的用户名和头像，点击可进入该用户的个人主页，点击列表中的取消关注按钮可取消对该用户的关注。

###### 我的粉丝

注册用户可查看我的粉丝列表，列表中展示当前用户粉丝的用户名和头像，点击可进入该用户的个人主页，点击列表中的关注按钮可关注该用户。

###### 用户设置

注册用户可对个人信息和账号进行设置。用户可设置头像、修改生日、性别以及自我介绍，用户可修改用户名和密码。

###### 搜索用户

在搜索框中输入用户名，点击搜索按钮或按下回车键，以字符串模糊匹配的方式对用户进行搜索，并将匹配的用户以列表的形式分页展示。

###### 查看用户主页

进入用户主页，可查看用户头像、用户名、性别、生日和自我介绍，可以点击按钮关注或取消关注用户。

###### 热搜榜

点击分享搜索框，将展示当前的热搜榜，包含热搜关键词和被搜索次数，点击热搜榜的关键词，将按照关键词进行搜索。热搜榜每天定时刷新清零，重新累计。

###### 密码加密

用户注册时输入的密码在存入数据库前进行MD5加密，避免用户密码以明文的形式存入数据库。

##### 软件技术、开发工具和开发环境

###### 前端

###### 后端

软件技术：SpringBoot、Spring、SpringMVC、mybatis、maven

开发工具：IntelliJ IDEA 2021.2、Navicat 15、Postman、git

开发环境：windows10、jdk1.8.0\_321、MySQL 8.0.27、Redis 5.0.14

#### 系统设计

##### 总体设计方案

###### 总体功能结构图



###### 总体设计方案说明

系统用户分为三种角色：访客用户、注册用户和管理员。系统的功能按照这三个角色进行划分。访客用户具有注册、主页、查看其他用户主页的功能。注册用户具有主页、登录、查看其他用户主页、分享互动、关注用户、用户后台、搜索用户的功能。管理员具有查看其他用户主页、搜索用户、注册用户管理、限制词管理的功能。

功能细分：

主页功能可细分为：分享列表、搜索分享、点赞榜、最新发布、查看分享、热搜榜。

分享互动功能可细分为：点赞分享、收藏分享、评论分享、回复评论。

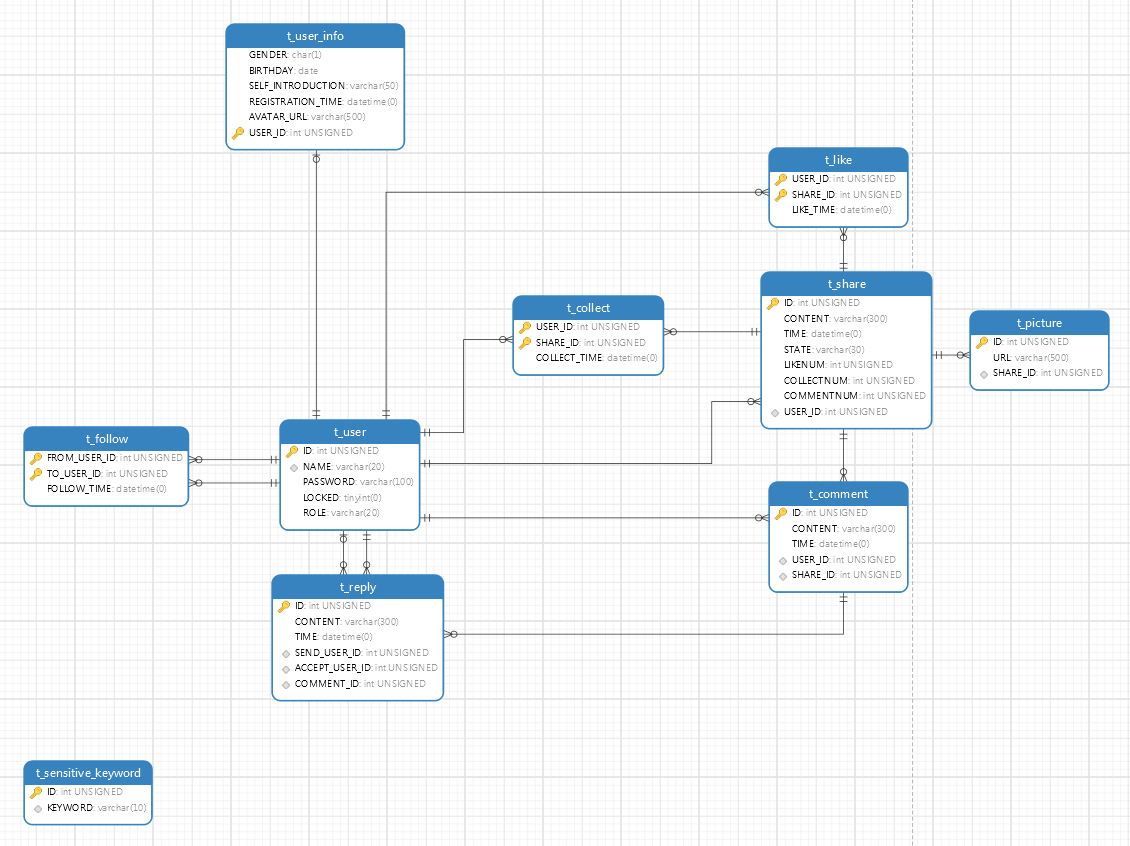
用户后台功能可细分为：我的分享、发布分享、修改分享、设置分享状态、我的收藏、我的关注、我的粉丝、用户设置。

##### 数据库设计

###### E-R图



###### 关系模式图



###### 字段说明

* + - 1. T\_USER用户表

ID 主键，用户ID

NAME 用户名

PASSWORD 密码

LOCKED 用户是否处于锁定状态

ROLE 用户角色

* + - 1. T\_USER\_INFO用户信息表

GENDER 性别

BIRTHDAY 生日

SELF\_INTRODUCTION 自我介绍

REGISTRATION\_TIME 注册时间

AVATAR\_URL 头像URL

USER\_ID 主键，外键，所属的用户ID

* + - 1. T\_SHARE分享表

ID 主键，分享ID

CONTENT 分享的内容

TIME 分享发布时间

STATE 分享的状态

LIKENUM 点赞数

COLLECTNUM 收藏数

COMMENTNUM 评论数

USER\_ID 外键，发布分享的用户ID

* + - 1. T\_COMMENT评论表

ID 主键，评论ID

CONTENT评论的内容

TIME 评论发布时间

USER\_ID 外键，发布评论的用户ID

SHARE\_ID 外键，评论所属的分享ID

* + - 1. T\_REPLY回复表

ID 主键，回复ID

CONTENT 回复的内容

TIME 回复发布时间

SEND\_USER\_ID 外键，发布回复的用户ID

ACCEPT\_USER\_ID 外键，接收回复的用户ID

COMMENT\_ID 外键，回复所属的评论ID

* + - 1. T\_SENSITIVE\_KEYWORD限制词表

ID 主键，限制词ID

KEYWORD 限制词

* + - 1. T\_PICTURE图片表

ID 主键，图片ID

URL 图片在文件系统中的路径

SHARE\_ID 外键，图片所属的分享ID

* + - 1. T\_FOLLOW关注表

FROM\_USER\_ID 主键，外键，发起关注的用户ID

TO\_USER\_ID 主键，外键，被关注的用户ID

FOLLOW\_TIME 关注的时间

* + - 1. T\_LIKE点赞表

USER\_ID 主键，外键，点赞用户ID

SHARE\_ID 主键，外键，获赞分享ID

LIKE\_TIME 点赞时间

* + - 1. T\_COLLECT收藏表

USER\_ID 主键，外键，收藏用户ID

SHARE\_ID 主键，外键，被收藏分享ID

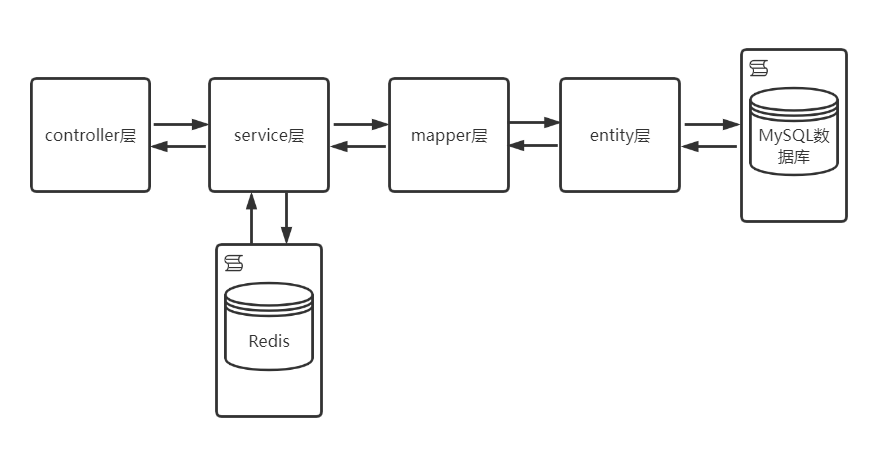
COLLECT\_TIME 收藏时间

##### 界面设计

##### 软件系统结构设计

###### 软件系统层次结构

* + - 1. 前端
      2. 后端



###### 各层次任务描述

* + - 1. 前端
      2. 后端
         1. controller层

存放暴露给前端的后端接口，接收前端请求，调用service层的业务逻辑，并将处理后获得的数据通过json的形式响应给前端。

* + - * 1. service层

存放业务逻辑处理，并通过调用mapper层的方法与数据库交互。

* + - * 1. mapper层

对数据库进行数据持久化操作，为service层提供数据库增删查改的方法。

* + - * 1. entity层

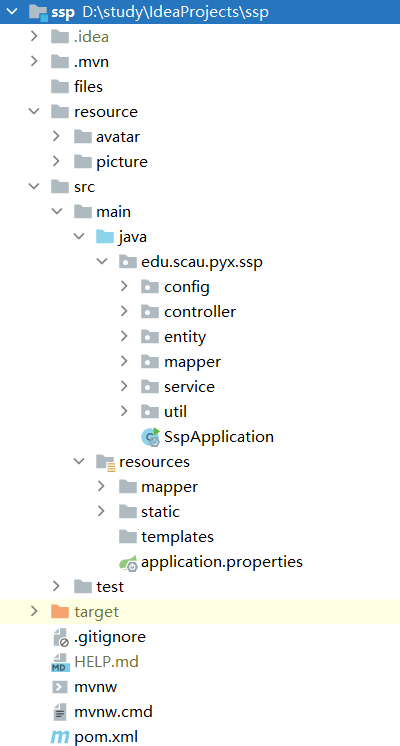
存放实体类，实体类中的属性值与数据库中的字段对应。获取数据库中的数据时，将字段映射成为对象的属性。

#### 系统实现

##### 系统项目结构

###### 前端

###### 后端



项目结构说明：

files文件夹是文件上传临时目录。由于在使用基于springboot的文件上传过程中，被上传的文件首先会被放入临时目录中，为避免出现临时目录不存在的情况，在files文件夹不存在时创建files文件夹，并指定files文件夹作为文件上传临时目录。

resource文件夹用于存放用户头像和分享图片，其中的avatar文件夹存放用户头像，picture文件夹存放分享图片。

src\main\java文件夹下：

config包存放配置类。

controller包存放controller类。

entity包存放entity类。

mapper包存放mapper类。

service包存放service类。

util包存放工具类。

src\main\resources文件夹下：

mapper文件夹存放mybatis映射文件。

static文件夹存放前端打包而成的静态文件。

##### 系统全局配置说明

###### 前端

###### 后端

MultipartConfig配置类，对文件上传临时目录进行配置，指定文件上传临时目录为项目路径下的files文件夹，当目录不存在时创建该目录。

MvcConfig配置类，配置允许跨域的路径为"/\*\*"，允许跨域的源为"\*"，允许的请求头为"\*"，允许的请求方法为"\*"，解决前后端不同源时的跨域问题。配置使用凭证为true，解决跨域时浏览器携带cookie的问题。

RedisConfig配置类，对RedisTemplate进行配置，配置后可注入RedisTemplate的实例。

application.properties文件中配置项目端口、配置MySQL数据库信息、向mybatis指定实体类所在包和映射文件所在目录、指定上传文件大小的最大值、配置Redis数据库信息。

##### 层次说明

###### 前端

###### 后端

* + - 1. controller层
         1. 实现的类

AdministratorController

CollectController

CommentController

FileController

FollowController

HotSearchController

LikeController

ReplyController

ShareController

UserController

* + - * 1. 实现方式说明

以CollectController为例：

@RestController  
@RequestMapping(value = "collect")  
public class CollectController {  
 @Autowired  
 private CollectService collectService;  
 @Autowired  
 private ShareService shareService;  
  
 @RequestMapping(value = "/add", method = RequestMethod.POST)  
 public boolean add(@RequestBody Collect collect,HttpSession session){  
 SystemUser user = (SystemUser) session.getAttribute("user");  
 if(user == null || !user.getRole().equals("user")){  
 return false;  
 }  
 collect.setUserId(user.getId());  
 return collectService.add(collect);  
 }  
  
 @RequestMapping(value = "/get", method = RequestMethod.GET)  
 public List<ShareListInfo> get(@RequestParam long begin, @RequestParam long length, HttpSession session){  
 SystemUser user = (SystemUser) session.getAttribute("user");  
 if(user == null || !user.getRole().equals("user")){  
 return null;  
 }  
 List<ShareListInfo> shareListInfoList = collectService.getCollectList(begin,length,user.getId());  
 shareService.setLikeAndCollectState(shareListInfoList, user.getId());  
 return shareListInfoList;  
 }  
  
 @RequestMapping(value = "/getcollectnum", method = RequestMethod.GET)  
 public long getCollectNum(HttpSession session){  
 SystemUser user = (SystemUser) session.getAttribute("user");  
 if(user == null || !user.getRole().equals("user")){  
 return 0;  
 }  
 return collectService.getCollectNum(user.getId());  
 }  
  
 @RequestMapping(value = "/cancel/{shareId}", method = RequestMethod.DELETE)  
 public boolean cancel(@PathVariable long shareId, HttpSession session){  
 SystemUser user = (SystemUser) session.getAttribute("user");  
 if(user == null || !user.getRole().equals("user")){  
 return false;  
 }  
 return collectService.cancel(shareId,user.getId());  
 }  
}

CollectController类存放收藏相关接口。由于系统采用前后端分离的方式进行开发，后端只需向前端返回json形式的数据，而不用控制页面的跳转，因此使用@RestController注解标注CollectController类。通过使用@RequestMapping标注类和方法配置controller方法的请求路径和请求方式。

add方法中：接收前端传递的collect对象。使用HttpSession对象获取当前登录用户的信息。若当前用户为空，即未登录，或者当前登录用户的角色不是”user”，则向前端返回添加收藏失败的信息。否则，将当前用户的ID放入collect对象中，调用并将collect对象传递给collectService对象中的add方法。向前端返回收藏是否成功的信息。

get方法中：接收前端传递的begin和length，这两个参数是分页信息，表示当前页起始的下标和长度。使用HttpSession对象获取当前登录用户的信息。若当前用户为空，即未登录，或者当前登录用户的角色不是”user”，则向前端返回空值。否则调用并将分页信息和当前用户ID传递给collectService对象中的getCollectList方法，获取收藏列表。调用并将收藏列表和用户ID传递给shareService对象中的setLikeAndCollectState方法，使收藏列表中的分享附带上当前用户对该分享的点赞和收藏状态信息。将收藏列表返回给前端。

getCollectNum方法中：使用HttpSession对象获取当前登录用户的信息。若当前用户为空，即未登录，或者当前登录用户的角色不是”user”，则向前端返回空的信息。否则调用并传递用户ID给collectService对象中的getCollectNum方法，获取并返回用户收藏的个数。

cancel方法中：接收前端传递的shareId，表示要取消收藏的分享的ID。使用HttpSession对象获取当前登录用户的信息。若当前用户为空，即未登录，或者当前登录用户的角色不是”user”，则向前端返回失败信息。否则调用并传递分享ID和用户ID给collectService对象中的cancel方法，并返回取消收藏的结果。

* + - 1. service层
         1. 实现的类

CollectServiceImpl

CommentServiceImpl

FileServiceImpl

FollowServiceImpl

HotSearchServiceImpl

LikeServiceImpl

PictureServiceImpl

ReplyServiceImpl

SensitiveKeywordServiceImpl

ShareServiceImpl

UserInfoServiceImpl

UserServiceImpl

* + - * 1. 实现方式说明

以HotSearchServiceImpl为例：

@Service  
public class HotSearchServiceImpl implements HotSearchService {  
 @Autowired  
 private RedisTemplate<String,String> redisTemplate;  
  
 @Override  
 public List<HotSearch> getList() {  
 List<HotSearch> hotSearchList = new ArrayList<>();  
 Set<String> hotSearchSet =

redisTemplate.opsForZSet().reverseRange("hotSearchSet", 0, 9);  
 assert hotSearchSet != null;  
 Iterator<String> iterator = hotSearchSet.iterator();  
 while (iterator.hasNext()){  
 String hotSearch = iterator.next();  
 long times =

redisTemplate.opsForZSet().score("hotSearchSet",hotSearch).longValue();  
 hotSearchList.add(new HotSearch(hotSearch,times));  
 }  
 return hotSearchList;  
 }  
}

HotSearchServiceImpl为热搜榜业务逻辑实现类。其中的getList方法为获取热搜榜，返回值为热搜榜链表。首先使用RedisTemplate类的实例redisTemplate获取key为hotSearchSet的sorted set的逆序前十的value。若值不为null，则遍历获取到的value，并使用redisTemplate获取该value对应的score。将value和score封装到HotSearch类的实例中，并将实例加入结果链表中，value和score分别表示热搜词和搜索次数。最后返回结果链表。

* + - 1. mapper层
         1. 实现的类

CollectMapper

CommentMapper

FollowMapper

LikeMapper

PictureMapper

ReplyMapper

SensitiveKeywordMapper

ShareMapper

UserInfoMapper

UserMapper

* + - * 1. 实现方式说明

以CollectMapper为例：

@Mapper  
public interface CollectMapper {  
 public boolean insert(Collect collect);  
  
 public List<ShareListInfo> getCollectList(long begin, long length, long userId);  
  
 public int isCollected(long shareId, long userId);  
  
 public boolean delete(long shareId, long userId);  
  
 public long getCollectNum(long id);  
}

<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE mapper  
 PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"

"http://mybatis.org/dtd/mybatis-3-mapper.dtd">  
<mapper namespace="edu.scau.pyx.ssp.mapper.CollectMapper">  
 <insert id="insert">  
 INSERT INTO T\_COLLECT( USER\_ID,SHARE\_ID )  
 VALUES ( #{userId},#{shareId} );  
 </insert>  
 <delete id="delete">  
 DELETE FROM T\_COLLECT  
 WHERE SHARE\_ID = #{shareId} AND USER\_ID = #{userId};  
 </delete>  
 <select id="getCollectList" resultType="edu.scau.pyx.ssp.entity.ShareListInfo">  
 SELECT S.ID AS id,  
 U.`NAME` AS username,  
 I.AVATAR\_URL AS userAvatarUrl,  
 S.TIME AS time,  
 S.CONTENT AS content,  
 S.STATE AS state,  
 S.LIKENUM AS likeNum,  
 S.COLLECTNUM AS collectNum,  
 S.COMMENTNUM AS commentNum  
 FROM T\_SHARE AS S,T\_USER AS U,T\_USER\_INFO AS I,T\_COLLECT AS C,T\_USER AS SU  
 WHERE S.USER\_ID=U.ID AND U.ID=I.USER\_ID AND C.SHARE\_ID=S.ID AND C.USER\_ID=#{userId} AND S.STATE!='private' AND S.STATE!='deleted' AND S.USER\_ID=SU.ID AND SU.LOCKED=0  
 ORDER BY C.COLLECT\_TIME DESC  
 LIMIT #{begin},#{length};  
 </select>  
 <select id="isCollected" resultType="int">  
 SELECT COUNT(\*)  
 FROM T\_COLLECT  
 WHERE USER\_ID = #{userId} AND SHARE\_ID = #{shareId};  
 </select>  
 <select id="getCollectNum" resultType="long">  
 SELECT COUNT(\*)  
 FROM T\_SHARE AS S,T\_USER AS U,T\_USER\_INFO AS I,T\_COLLECT AS C,T\_USER AS SU  
 WHERE S.USER\_ID=U.ID AND U.ID=I.USER\_ID AND C.SHARE\_ID=S.ID AND C.USER\_ID=#{userId} AND S.STATE!='private' AND S.STATE!='deleted' AND S.USER\_ID=SU.ID AND SU.LOCKED=0  
 </select>  
</mapper>

在mybatis映射文件中编写SQL语句实现mapper接口中的方法。

insert方法：

INSERT INTO T\_COLLECT( USER\_ID,SHARE\_ID )  
 VALUES ( #{userId},#{shareId} );

一条插入语句，将collect实例中获取的userId和shareId插入T\_COLLECT表中。

getCollectList方法：

SELECT S.ID AS id,  
 U.`NAME` AS username,  
 I.AVATAR\_URL AS userAvatarUrl,  
 S.TIME AS time,  
 S.CONTENT AS content,  
 S.STATE AS state,  
 S.LIKENUM AS likeNum,  
 S.COLLECTNUM AS collectNum,  
 S.COMMENTNUM AS commentNum  
 FROM T\_SHARE AS S,T\_USER AS U,T\_USER\_INFO AS I,T\_COLLECT AS C,T\_USER AS SU  
 WHERE S.USER\_ID=U.ID AND U.ID=I.USER\_ID AND C.SHARE\_ID=S.ID AND C.USER\_ID=#{userId} AND S.STATE!='private' AND S.STATE!='deleted' AND S.USER\_ID=SU.ID AND SU.LOCKED=0  
 ORDER BY C.COLLECT\_TIME DESC  
 LIMIT #{begin},#{length};

对T\_SHARE、T\_USER、T\_USER\_INFO、T\_COLLECT表的联合查询。查询条件为收藏用户为当前用户，分享未设置为私有或被删除，发布分享的用户未被锁定。查询结果按照时间降序排列。查询结果行偏移量为begin，最大length个。

isCollected方法：

SELECT COUNT(\*)  
 FROM T\_COLLECT  
 WHERE USER\_ID = #{userId} AND SHARE\_ID = #{shareId};

从T\_COLLECT表中查询用户ID为userId、分享ID为shareId的记录数。

delete方法：

DELETE FROM T\_COLLECT  
 WHERE SHARE\_ID = #{shareId} AND USER\_ID = #{userId};

从T\_COLLECT表中删除用户ID为userId、分享ID为shareId的记录。

getCollectNum方法：

SELECT COUNT(\*)  
 FROM T\_SHARE AS S,T\_COLLECT AS C,T\_USER AS SU  
 WHERE C.SHARE\_ID=S.ID AND C.USER\_ID=#{userId} AND S.STATE!='private' AND S.STATE!='deleted' AND S.USER\_ID=SU.ID AND SU.LOCKED=0

对T\_SHARE、T\_COLLECT、T\_USER表联合查询，查询条件为收藏用户ID为userId、分享未被删除或设置为私有、发布分享的用户未被锁定，查询结果为符合条件的记录数。

* + - 1. entity层
         1. 实现的类

Administrator

Collect

Comment

CommentListInfo

Follow

HotSearch

Like

Picture

Reply

ReplyListInfo

Role

SensitiveKeyword

Share

ShareListInfo

SystemUser

UserInfo

UserListInfo

* + - * 1. 实现方式说明

以Share为例：

public class Share {  
 private long id;  
 private String content;  
 private String state;  
 private long likeNum;  
 private long collectNum;  
 private long commentNum;  
 private long userId;  
  
 public Share() {  
 }  
  
 public Share(long id, String content, String state, long likeNum, long collectNum, long commentNum, long userId) {  
 this.id = id;  
 this.content = content;  
 this.state = state;  
 this.likeNum = likeNum;  
 this.collectNum = collectNum;  
 this.commentNum = commentNum;  
 this.userId = userId;  
 }  
  
 public long getId() {  
 return id;  
 }  
  
 public void setId(long id) {  
 this.id = id;  
 }  
  
 public String getContent() {  
 return content;  
 }  
  
 public void setContent(String content) {  
 this.content = content;  
 }  
  
 public String getState() {  
 return state;  
 }  
  
 public void setState(String state) {  
 this.state = state;  
 }  
  
 public long getLikeNum() {  
 return likeNum;  
 }  
  
 public void setLikeNum(long likeNum) {  
 this.likeNum = likeNum;  
 }  
  
 public long getCollectNum() {  
 return collectNum;  
 }  
  
 public void setCollectNum(long collectNum) {  
 this.collectNum = collectNum;  
 }  
  
 public long getCommentNum() {  
 return commentNum;  
 }  
  
 public void setCommentNum(long commentNum) {  
 this.commentNum = commentNum;  
 }  
  
 public long getUserId() {  
 return userId;  
 }  
  
 public void setUserId(long userId) {  
 this.userId = userId;  
 }  
}

属性均使用private修饰，向外界提供public修饰的无参构造方法、有参数的构造方法和属性的getter和setter方法。

#### 软件测试

#### 项目总结

##### 项目实现的功能

###### 显示分享列表

以分页的形式显示分享列表。列表中的每个分享要展示分享的内容、图片，发布分享用户的用户名、头像，分享发布的时间，分享的点赞数、评论数和收藏数，以及当前用户是否已点赞收藏该分享。点击分享内容跳转到分享详情页面，点击分享图片展示大图，点击用户头像可跳转到用户主页。根据用户是否已点赞、收藏该分享，点赞按钮、收藏按钮以不同颜色显示。点击评论按钮可展示用户评论列表，再次点击可将用户评论列表折叠。点击评论中的查看回复按钮可查看该评论下的回复。

###### 搜索分享

在搜索框中输入内容，点击搜索按钮或按下回车键，以字符串模糊匹配的方式对分享的内容进行搜索，并将匹配的分享以分享列表的形式分页展示。

###### 点赞榜

按照点赞数降序展示分享列表。

###### 最新发布

按照发布时间最新的顺序展示分享列表。

###### 查看分享

点击分享列表中分享的内容进入分享详细页面。

###### 登录/注册

点击注册按钮进入注册页面，用户输入用户名、密码、确认密码，若用户名重复或确认密码与密码不匹配，则不允许注册，若合法，则点击注册按钮进行注册。点击登录按钮进入登录页面，输入用户名和密码进行登录，用户名存在且密码正确则登录成功。

###### 点赞分享

注册用户可点赞分享。分享的点赞按钮根据用户是否已点赞以不同的颜色展示。若未点赞，点击点赞按钮对分享点赞，若已点赞，点击点赞按钮对分享取消点赞。

###### 收藏分享

注册用户可收藏分享。分享的收藏按钮根据用户是否已收藏以不同的颜色展示。若未收藏，点击收藏按钮收藏分享，若已收藏，点击收藏按钮对分享取消收藏。

###### 评论分享

注册用户可评论分享。用户点击分享的评论按钮后，展示评论输入框，用户在输入框中输入评论内容，点击发表按钮发表评论。新发表的评论实时展示在评论列表中。每条评论需要展示出评论的内容，发布评论的用户的用户名、头像，评论发布时间以及回复数。

###### 回复评论

注册用户可回复评论。用户点击评论中的回复按钮后，展示回复输入框，用户在输入框中输入回复内容，点击提交按钮提交回复。实时更新评论的回复状态。每条回复需要展示出回复的内容，提交回复的用户的用户名、头像，回复提交时间。

注册用户可对回复进行回复。用户点击回复中的回复按钮进行回复。与对评论进行回复不同的是，对回复的回复在回复的内容前会显示“回复@xxx”，以明确所回复的用户。

###### 我的分享

以分页分享列表的形式展示当前用户发布的分享。包含分享内容、用户名、用户头像、发布时间信息。点击详情进入分享详情页面，点击编辑按钮展示分享编辑页面、点击删除按钮删除已发布的分享。

###### 发布分享

注册用户可发布分享。进入分享发布页面，可输入分享内容、设置分享是否公开、以及上传图片，已选择的图片将会展示出来，每个分享最多可有6个图片。点击发布按钮发布分享。

###### 修改分享

在分享修改页面中，用户可对分享的内容、分享是否公开、以及分享的图片进行编辑，点击提交按钮确认修改。

###### 分享状态

分享的状态分为“仅自己可见”和“公开”，在发布分享时可进行设置，在分享发布后，可在修改分享页面中进行修改。

###### 我的收藏

以分页分享列表的形式展示当前用户收藏的分享。包含分享内容、用户名、用户头像、发布时间信息。点击详情进入分享详情页面，点击删除按钮可取消收藏。

###### 注册用户管理

管理员可以禁⽌或重新允许⼀个注册⽤户进⾏分享活动。处于禁⽌状态的注册⽤户的分享均不可⻅。

###### 限制词管理

管理员可对限制词进行管理，可添加或删除限制词。

###### 限制词检测

在用户发布分享或修改分享内容前，对分享的内容进行限制词检测，若分享内容中存在限制词，则不允许用户发布或修改分享内容。

###### 关注用户

注册用户可关注用户。进入其他用户的主页，根据用户是否已关注，关注按钮显示为“关注”或“取消关注”，点击按钮可关注用户或取消关注用户。

###### 我的关注

注册用户可查看我的关注列表，列表中展示当前用户已关注用户的用户名和头像，点击可进入该用户的个人主页，点击列表中的取消关注按钮可取消对该用户的关注。

###### 我的粉丝

注册用户可查看我的粉丝列表，列表中展示当前用户粉丝的用户名和头像，点击可进入该用户的个人主页，点击列表中的关注按钮可关注该用户。

###### 用户设置

注册用户可对个人信息和账号进行设置。用户可设置头像、修改生日、性别以及自我介绍，用户可修改用户名和密码。

###### 搜索用户

在搜索框中输入用户名，点击搜索按钮或按下回车键，以字符串模糊匹配的方式对用户进行搜索，并将匹配的用户以列表的形式分页展示。

###### 查看用户主页

进入用户主页，可查看用户头像、用户名、性别、生日和自我介绍，可以点击按钮关注或取消关注用户。

###### 热搜榜

点击分享搜索框，将展示当前的热搜榜，包含热搜关键词和被搜索次数，点击热搜榜的关键词，将按照关键词进行搜索。热搜榜每天定时刷新清零，重新累计。

###### 密码加密

用户注册时输入的密码在存入数据库前进行MD5加密，避免用户密码以明文的形式存入数据库。

##### 改进空间

项目在安全性方面存在改进空间。由于未使用安全框架，用户的验证、授权均使用session实现。这种方式安全性不高，由于session是依赖于cookie实现的，这就存在跨站点请求伪造的风险。并且在controller中需要从session获取用户信息，然后对用户是否已登录以及登录用户的权限信息进行判断，验证授权的耦合度较高，并且代码冗余较多。

##### 个人心得体会

###### 彭翊轩

在本次现代软件开发技术期末考核项目中负责后端。本次的项目采用前后端分离的架构模式。相比于以往的web项目，前后端耦合度更低，后端能够更专注于对后台数据的处理。对于前端的页面和渲染，后端不需要过度关注，只需要根据接口文档制定的接口规范进行开发，通过暴露接口的方式为前端提供数据。后端采用的是分层的体系结构，通过分层达到后端各层的高内聚低耦合。其中，controller层存放暴露给前端的后端接口，接收前端请求，调用service层的业务逻辑，并将处理后获得的数据通过json的形式响应给前端。service层存放业务逻辑处理，并通过调用mapper层的方法与数据库交互。mapper层对数据库进行数据持久化操作，为service层提供数据库增删查改的方法。entity层存放实体类，实体类中的属性值与数据库中的字段对应。获取数据库中的数据时，将字段映射成为对象的属性。数据库设计是后端开发的关键，根据需求设计出良好的数据库，才能在整个后端开发过程中更高效地处理数据，并为后续的维护和拓展提供便利。