**华南理工大学广州学院**

**2021-2022学年度第1学期**

**课程名称： Java EE框架应用开发项目实践**

**题　　目： 城理人论坛的设计与实现**

**专业班级： 2019级软件工程6班**

**目录**

[1、系统设计 3](#_Toc88564313)

[1.1、系统功能需求 3](#_Toc88564314)

[1.2、系统模块划分 3](#_Toc88564315)

[2、数据库设计 5](#_Toc88564316)

[2.1、数据库概念结构设计 5](#_Toc88564317)

[2.2、数据库逻辑结构设计 6](#_Toc88564318)

[3、系统管理 8](#_Toc88564319)

[3.1、技术选项与包管理 8](#_Toc88564320)

[3.2、页面管理 9](#_Toc88564321)

[3.3、应用目录结构与项目层次 17](#_Toc88564322)

[3.4、配置文件管理 18](#_Toc88564323)

[4、接口设计 18](#_Toc88564324)

[4.1、一般接口的设计 18](#_Toc88564325)

[4.2、详细接口文档 19](#_Toc88564326)

[5、组件设计 30](#_Toc88564327)

[5.1、权限验证模块设计 30](#_Toc88564328)

[5.2、功能模块详细设计 30](#_Toc88564329)

[6、系统实现 30](#_Toc88564330)

[6.1、关键代码分析 30](#_Toc88564331)

[6.2、疑难问题解决 32](#_Toc88564332)

[7、项目的发布和测试 32](#_Toc88564333)

[7.1、项目发布过程 32](#_Toc88564334)

[7.2、项目测试过程 32](#_Toc88564335)

[8、分析程序的优点和不足,遇到的困难及解决的问题,总结自己的收获 33](#_Toc88564336)

[9、参考目录,写出自己编写该程序时阅读的书籍和出处 33](#_Toc88564337)

**1、系统设计**

**1.1、系统功能需求**

城理人论坛为城理学生提供了一个在线的交流平台,能够让那些对同一个话题感兴趣的城理学子聚在一起,方便地展开交流和互相帮助。为城理学子提供一个表达和交流思想的自由网络空间,并以此汇聚志同道合的网友。

一个典型的论坛一般都需要提供良好的维护页面,即管理人员可以借助后台维护管理的页面实现对论坛内容实施更新维护。从前台功能上来看需要包括帐号密码登陆、论坛板块的分类显示、对应板块的帖子显示、帖子详细内容显示等。同时也应该为论坛提供发布帖子,回复帖子、删除帖子等功能。

1)登陆帐号

打开论坛小程序,界面上显示帐号密码的输入框,以及帐号类型的单选框,该界面能够根

据数据库中存放的帐号信息进行判断,然后返回给用户结果。

2)论坛板块的分类显示

进入论坛中心主页,页面上应该能够根据数据库中存放的信息分类显示出最新的板块内容。每个论坛的模板都会显示其对应的帖子数与热度,用户可以根据自己的喜好进行点击操作。用户可以点击感兴趣的板块,就能够跳转到有关该类型的帖子页面,让用户能够清楚的看到所有相关类型的帖子,每个帖子也都会显示对应的回帖数与热度。

3)帖子详细内容

用户点击感兴趣的帖子后,能够查看到该帖子的详细内容,比如:论坛的正文、网友们的回帖内容,并且用户也可以对该帖子发表自己的看法与观点。

4)论坛后台管理功能

论坛的超级管理员可以根据客户的需求随时对后台数据库进行增、删、改等功能,例如,超级管理员可以在数据库中添加最新的板块,还可以随时删除某些不合规的板块及相应的帖子内容,以及对一些原有的论坛帖子做必要的修改。

如图1所示显示了该系统的功能,它包括两大模块,分别是帖子信息浏览模块和后台管理模块。

其中,帖子信息浏览模块主要完成板块分类的显示,能够让用户一目了然的了解到自己感兴趣的帖子,并且能够查看帖子的详细内容与回复内容。后台管理模块主要完成对论坛的维护,其中就有增加、修改、删除、封号等功能。

**1.2、系统模块划分**

1)帖子信息浏览模块

帖子信息浏览模块主要包括论坛分类显示、详细帖子显示及帖子对应的所有回复,如图2所示。

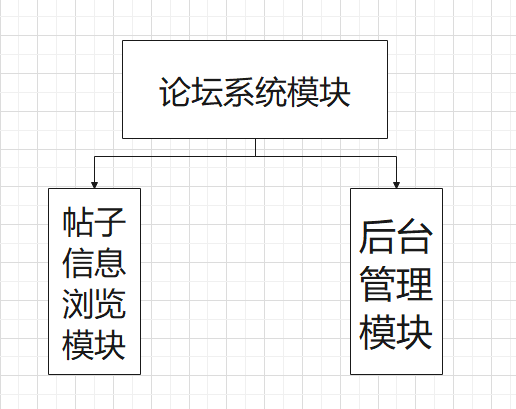
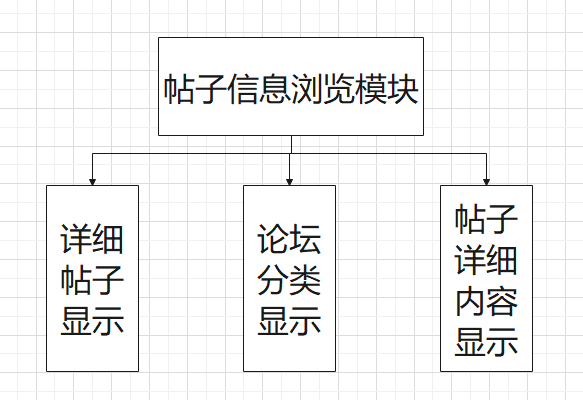
 

图1 论坛系统功能模块 图2 系统浏览模块

a)论坛板块分类显示

该模块实现了论坛板块的分类显示,例如,将所有帖子分为编程区、学习区和恋爱区等类别,分类显示各种帖子以便用户选择感兴趣的板块进行详细帖子的阅读。

b)详细帖子显示

依据用户所选择的板块显示对应的相关帖子。

c)相关帖子内容显示

依据用户所选择的帖子,向用户显示对应帖子的详细内容。

2)后台管理模块

后台管理模块包含管理论坛板块、管理每个板块的帖子对其进行添加、修改和删除,以及管理版主和普通用户对其进行权限的管理,如图3所示。

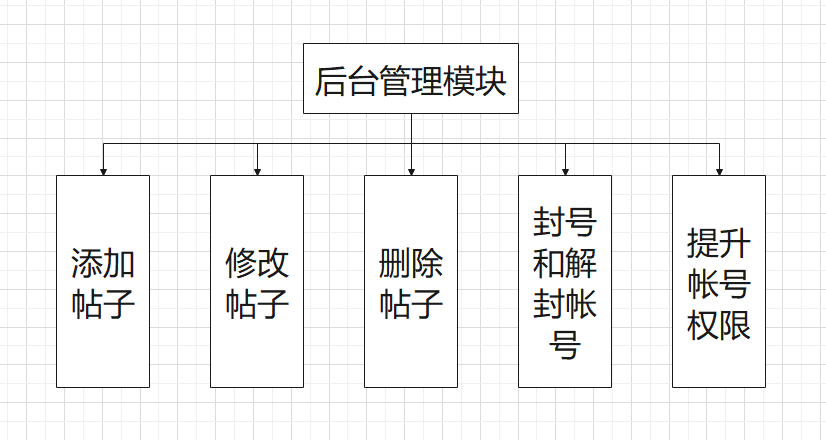


图3 后台管理模块

* 1. 添加帖子

添加帖子主要添加一些有意义的帖子到对应的模块。

* 1. 修改帖子

修改帖子主要负责对现有帖子进行修改。

* 1. 删除帖子

删除帖子主要负责对不合规的帖子进行删除操作。

* 1. 封号和解封帐号

封号主要是对违规的帐号进行封禁,假如号主申诉解封成功时,对其帐号进行解封操作。

* 1. 提升帐号权限

提升帐号权限主要是将普通用户提升为版主,对其所在的板块进行管理,便于管理员对整个论坛进行管理。

**2、数据库设计**

**2.1、数据库概念结构设计**

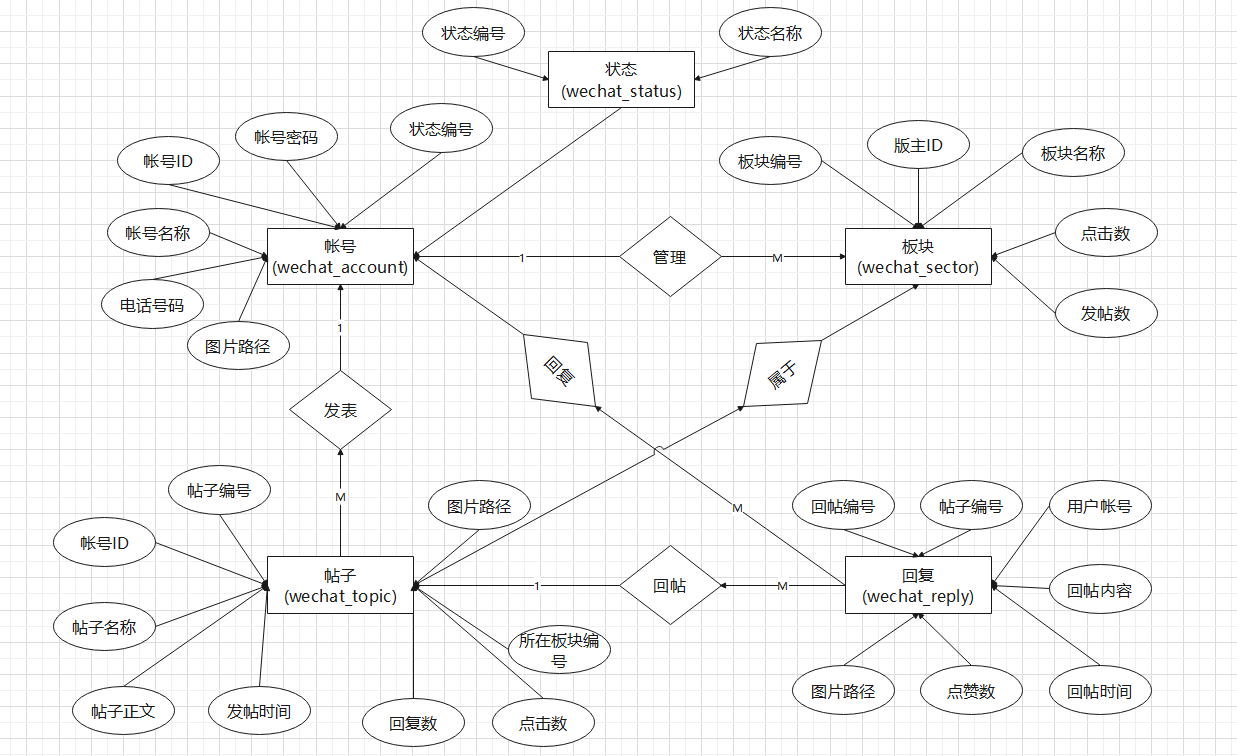


图2.1 E-R图

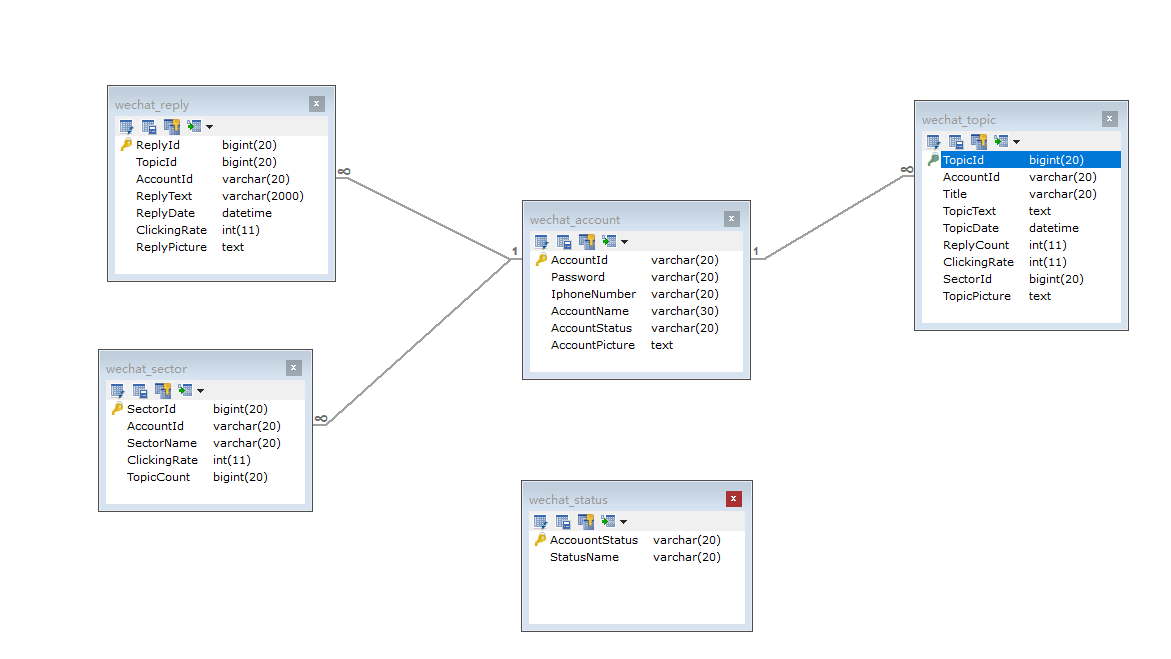


图2.2 数据库数据模型图

**2.2、数据库逻辑结构设计**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 表名称 | Wechat\_account | | | | |
| 列名 | 数据类型 | 主键 | 非空 | 自增 | 说明 |
| AccountId | Varchar | Y | Y | N | 用户帐号 |
| Password | Varchar | N | Y | N | 用户密码 |
| IphoneNumber | Varchar | N | Y | N | 用户手机号 |
| AccountName | Varchar | N | Y | N | 用户名字 |
| AccountStatus | Varchar | N | Y | N | 帐号状态 |

**表1 用户登陆表的逻辑结构设计**

**表2 帖子回复的逻辑结构设计**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 表名称 | Wechat\_reply | | | |  |
| 列名 | 数据类型 | 主键 | 非空 | 自增 | 说明 |
| ReplyId | Varchar | Y | Y | Y | 回帖编号 |
| TopicId | Bigint | N | Y | N | 帖子编号 |
| AccountId | Varchar | N | Y | N | 用户帐号 |
| ReplyText | Varchar | N | Y | N | 回帖内容 |
| ReplyDate | Datetime | N | N | N | 回帖时间 |
| ClickingRate | Int | N | N | N | 点击数 |

**表3 论坛板块的逻辑结构设计**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 表名称 | Wechat\_sector | | | | |
| 列名 | 数据类型 | 主键 | 非空 | 自增 | 说明 |
| SectorId | Bigint | Y | Y | Y | 板块编号 |
| AccountId | Varchar | N | Y | N | 版主帐号 |
| SectorName | Varchar | N | Y | N | 板块名字 |
| ClickingRate | Int | N | N | N | 点击数 |
| TopicCount | Bigint | N | N | N | 发帖数 |

**表4 用户类型的逻辑结构设计**

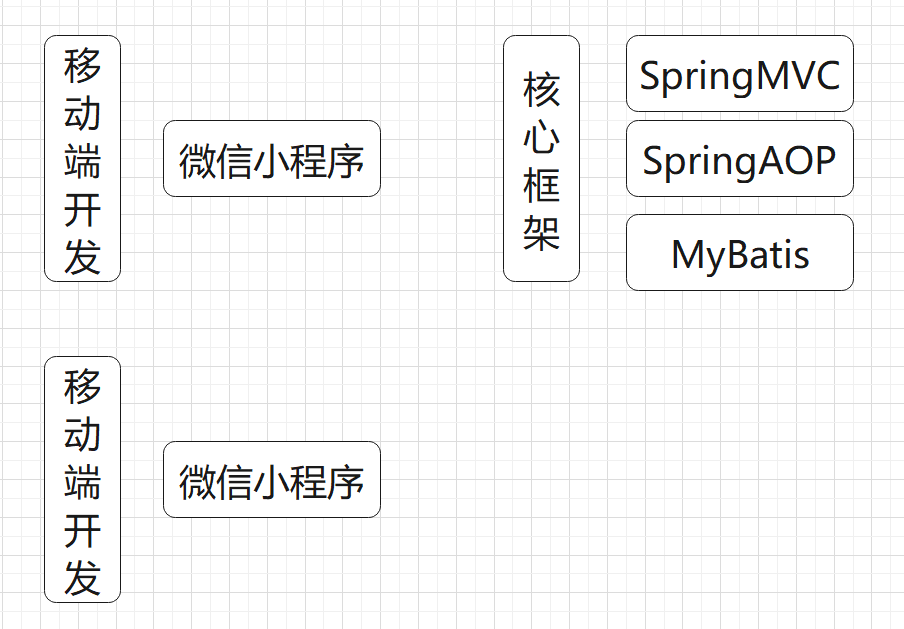
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 表名称 | Wechat\_status | | | | |
| 列名 | 数据类型 | 主键 | 非空 | 自增 | 说明 |
| AccountStatus | Varchar | Y | Y | N | 用户等级 |
| StatusName | Varchar | N | Y | N | 等级名字 |

**表5 帖子的逻辑结构设计**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 表名称 | Wechat\_topic | | | | |
| 列名 | 数据类型 | 主键 | 非空 | 自增 | 说明 |
| TopicId | Bigint | Y | Y | Y | 帖子编号 |
| AccountId | Varchar | N | Y | N | 用户帐号 |
| Title | Varchar | N | Y | N | 帖子标题 |
| TopicText | Text | N | Y | N | 帖子正文 |
| TopicDate | Datetime | N | Y | N | 发帖时间 |
| ReplyCount | Int | N | N | N | 回复数 |
| ClickingRate | Int | N | N | N | 点击数 |
| SectorId | Bigint | N | Y | N | 板块编号 |

**3、系统管理**

**3.1、技术选项与包管理**



**3.2、页面管理**



图1：登录界面

小程序登录界面，用户或管理员输入号码和密码后，有单选按钮“用户”和“管理员”按钮选择点击登录。

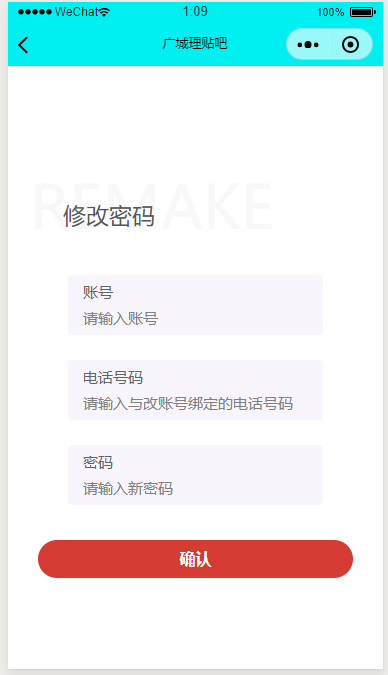


图2:修改密码界面

修改密码界面，用户输入注册时账号和电话号码、和新密码，只有当账号和电话号码和注册时一样是方可修改密码。



图3：注册页面

用户输入账号，密码，电话号码，用户名，然后添加头像图片，点击注册便可注册一个新账号，账号重复或者没添加头像注册不了



图4：管理员的首页

管理员可在这里对用户进行封号，解封，升级为版主，除去版主，新增板块，删除板块功能



图5:管理员社区界面

管理员在这个界面进行删帖，左上角黑色按钮是发帖，右上角是模糊搜索功能，中间部分是帖子，管理员可以对帖子进行删除，或者点击进帖子进行回复，看帖。



图6：帖子界面

如图，有帖主模块和读者模块，读者可在下方回复，加号为添加图片



图7：点亮回复

用户点击回复下面的赞后（亮了）值会加1



图8：删除回复

该功能是由管理员独有，版主也不能拥有此功能



图9：发帖界面

用户可在此界面进行发帖，可发无图帖和有图帖，有图帖选择加号添加图片再选择区域即可发送成功



图10：模糊搜索界面

用户可在此界面查询帖子，在最上方随意输入帖子名，点击搜索即可查询到相关帖子。

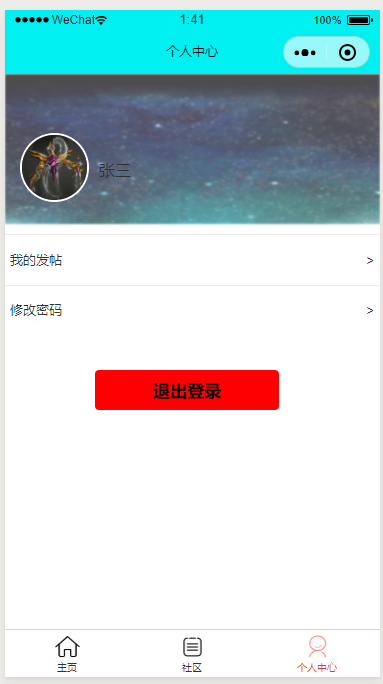


图11：个人界面

用户在此界面显示头像，可查看自己的发帖记录，也可修改密码



图12：我的发帖界面

用户可在此查看自己的发帖记录，不论是普通用户还是版主还是管理员都可在此界面删除自己发过的帖子



图13:版主的社区界面(该版主目前是编程区版主)：

版主可在此界面删除自己管理区域的帖子



图14：编程区模块（当前的用户是编程区版主）：

下方的删除按钮是只有管理员或者编程区版主才会显示出来，点击即可删除，恋爱区同理



图15:点击删除功能：

会弹出提示，点击确认即可删除帖子成功



图16:版主和普通用户的首页界面

版主和普通用户可在此界面查看该贴吧的吧规，点击上面的轮播图或者下面的本站规定都可以查看

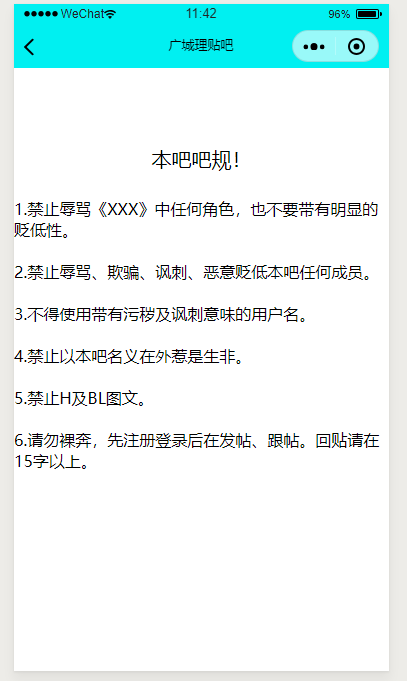


图17：吧规图

该页面展示该贴吧的吧规

**3.3、应用目录结构与项目层次**

前端：

├─images 存放图片的根目录

│ ├─alltie 社区界面需要的图片目录

│ ├─index 首页界面需要的图片目录

│ │ └─轮播图 轮播图里边需要图片的目录

│ ├─personal 个人页面需要的图片目录

│ ├─send 发帖需要的图片目录

│ ├─tabBar tabBar需要的图片目录

│ └─tie 帖子内容需要的图片目录

├─pages 界面的根目录

│ ├─alltie “社区界面”界面

│ ├─bagui 吧规界面

│ ├─banzhu 版主规则界面

│ ├─bshuitie 水贴规则界面

│ ├─forget “修改密码”界面

│ ├─index “首页”界面

│ ├─login “登录”界面

│ ├─mytie “我的帖子”界面

│ ├─personal “个人中心”界面

│ ├─register “注册”界面

│ ├─search “模糊查询”界面

│ ├─send “发帖”界面

│ └─tie “帖子”界面

└─utils 配置端口文件

后端:

bigWork

├─.idea

│ ├─artifacts

│ └─libraries

├─.settings

├─build

│

├─lib

├─src

│ ├─controller //控制器包,系统的控制器类都在这个包

│ │ ├─error

│ │ └─user

│ ├─dao //Mapper接口包,对应着mapperxml的操作id

│ ├─interceptor

│ ├─mybatis

│ │ └─mapper //XXXXmapper.xml文件夹

│ ├─pojo //POJO包 持久类存放

│ └─service //service实现类包

└─WebContent

├─META-INF

└─WEB-INF //spring-mybatis整合配置文件在这里

├─lib

└─views

**3.4、配置文件管理**

配置文件为springMVC-servlet配置文件以及web配置文件。

**4、接口设计**

**4.1、一般接口的设计**

a) 先写实体类entity，定义对象的属性，（可以参照数据库中表的字段来设置，数据库的设计应该在所有编码开始之前）。

b)写Mapper.xml（Mybatis），其中定义的功能，对应要对数据库进行的那些操作，比如 insert、selectAll、selectByKey、delete、update等。

c) 写Mapper.java，将Mapper.xml中的操作按照id映射成Java函数。

d) 写Service.java，为控制层提供服务，接受控制层的参数，完成相应的功能，并返回给控制层。

e) 写Controller.java，连接页面请求和服务层，获取页面请求的参数，通过自动装配，映射不同的URL到相应的处理函数，并获取参数，对参数进行处理，之后传给服务层。

**4.2、详细接口文档**

1)登陆帐号

|  |  |  |  |
| --- | --- | --- | --- |
| 登陆帐号 | | | |
| 接口描述 | 登陆帐号 | | |
| URL | /account/login | | |
| 请求类型 | post | | |
| 请求参数名 | | 数据类型 | 说明 |
| accountId | | String | 帐号 |
| password | | String | 密码 |
| 返回属性名 | | 数据类型 | 说明 |
| 1.err\_code | | Integer |  |
| 2.err\_msg | | String | 获取失败原因 |
| 3.data | | String | 返回对应帐号状态 |

2)注册帐号

|  |  |  |  |
| --- | --- | --- | --- |
| 注册帐号 | | | |
| 接口描述 | 注册帐号 | | |
| URL | /account/add | | |
| 请求类型 | post | | |
| 参数名 | | 数据类型 | 说明 |
| accountId | | String | 帐号 |
| password | | String | 密码 |
| iphoneNumber | | String | 电话号码 |
| accountName | | String | 帐号名字 |
| accountPicture | | Text | 图片路径 |
| 返回属性名 | | 数据类型 | 说明 |
| 1.err\_code | | Integer |  |
| 2.err\_msg | | String | 获取失败原因 |
| 3.data | | String | 返回1即为成功,0即为失败 |

3)找回/修改密码

|  |  |  |  |
| --- | --- | --- | --- |
| 找回/修改密码 | | | |
| 接口描述 | 找回/修改密码 | | |
| URL | /account/update | | |
| 请求类型 | post | | |
| 参数名 | | 数据类型 | 说明 |
| accountId | | String | 帐号 |
| password | | String | 新密码 |
| iphoneNumber | | String | 电话号码 |
| 返回属性名 | | 数据类型 | 说明 |
| 1.err\_code | | Integer |  |
| 2.err\_msg | | String | 获取失败原因 |
| 3.data | | String | 返回1即为成功,0即为失败 |

4)查看用户自己所有的帖子

|  |  |  |  |
| --- | --- | --- | --- |
| 查看用户自己所有的帖子 | | | |
| 接口描述 | 查看用户自己所有的帖子 | | |
| URL | /topic/list | | |
| 请求类型 | get | | |
| 参数名 | | 数据类型 | 说明 |
| accountId | | String | 帐号 |
| 返回属性名 | | 数据类型 | 说明 |
| 1.err\_code | | Integer |  |
| 2.err\_msg | | String | 获取失败原因 |
| 3.data | | Object:list<topicInfo> |  |
| 3.1.title | | String | 帖子标题 |
| 3.2.topicText | | String | 帖子正文 |
| 3.3.topicDate | | String | 发帖日期 |
| 3.4.replyCount | | Integer | 回复数 |
| 3.5.clickingRate | | Integer | 点击数 |
| 3.6.sectorName | | String | 所在板块名字 |
| 3.6.topicPicture | | String | 图片路径 |

5)查看所有的帖子

|  |  |  |  |
| --- | --- | --- | --- |
| 查看所有的帖子 | | | |
| 接口描述 | 查看所有的帖子 | | |
| URL | /topic/allList | | |
| 请求类型 | get | | |
| 返回属性名 | | 数据类型 | 说明 |
| 1.err\_code | | Integer |  |
| 2.err\_msg | | String | 获取失败原因 |
| 3.data | | Object:list<topicInfo> |  |
| 3.1.title | | String | 帖子标题 |
| 3.2.topicText | | String | 帖子正文 |
| 3.3.topicDate | | String | 发帖日期 |
| 3.4.replyCount | | Integer | 回复数 |
| 3.5.clickingRate | | Integer | 点击数 |
| 3.6.sectorName | | String | 所在板块名字 |
| 3.7.accountPicture | | String | 图片路径 |

6)用户发帖

|  |  |  |  |
| --- | --- | --- | --- |
| 用户发帖 | | | |
| 接口描述 | 用户发帖 | | |
| URL | /topic/add | | |
| 请求类型 | post | | |
| 参数名 | | 数据类型 | 说明 |
| accountId | | String | 用户帐号 |
| title | | String | 帖子标题 |
| topicText | | String | 帖子正文 |
| sectorId | | Integer | 板块编号 |
| topicPicture | | String | 图片路径 |
| 返回属性名 | | 数据类型 | 说明 |
| 1.err\_code | | Integer |  |
| 2.err\_msg | | String | 获取失败原因 |
| 3.data | | String | 返回1表示成功,0即为失败 |

7)回复帖子

|  |  |  |  |
| --- | --- | --- | --- |
| 回复帖子 | | | |
| 接口描述 | 回复帖子 | | |
| URL | /reply/add | | |
| 请求类型 | post | | |
| 参数名 | | 数据类型 | 说明 |
| accountId | | String | 帐号 |
| topicId | | Integer | 帖子编号 |
| replyText | | String | 回复内容 |
| replyPicture | | String | 图片路径 |
| 返回属性名 | | 数据类型 | 说明 |
| 1.err\_code | | Integer |  |
| 2.err\_msg | | String | 获取失败原因 |
| 3.data | | String | 返回1即为成功,0即为失败 |

8)查看对应帖子的回复内容

|  |  |  |  |
| --- | --- | --- | --- |
| 查看对应帖子的回复内容 | | | |
| 接口描述 | 查看对应帖子的回复内容 | | |
| URL | /reply/list | | |
| 请求类型 | post | | |
| 参数名 | | 数据类型 | 说明 |
| topicId | | Integer | 帖子编号 |
| 返回属性名 | | 数据类型 | 说明 |
| 1.err\_code | | Integer |  |
| 2.err\_msg | | String | 获取失败原因 |
| 3.data | | Object:list<ReplyInfo> |  |
| 3.1.accountId | | String | 帐号 |
| 3.2.accountName | | String | 帐号名字 |
| 3.3.replyText | | String | 回复内容 |
| 3.4.replyDate | | String | 回复时间 |
| 3.5.clickingRate | | Integer | 回复点击数 |
| 3.6.replyPicture | | String | 图片路径 |

9)版主查看所管理的模块

|  |  |  |  |
| --- | --- | --- | --- |
| 版主查看所管理的模块 | | | |
| 接口描述 | 版主查看所管理的模块 | | |
| URL | /sector/list | | |
| 请求类型 | post | | |
| 参数名 | | 数据类型 | 说明 |
| accountId | | String | 版主帐号 |
| 返回属性名 | | 数据类型 | 说明 |
| 1.err\_code | | Integer |  |
| 2.err\_msg | | String | 获取失败原因 |
| 3.data | | Object:list<SectorInfo> |  |
| 3.1.sectorId | | Integer | 板块编号 |
| 3.2.sectorName | | String | 板块名字 |
| 3.3.clickingRate | | Integer | 板块点击数 |
| 3.4.topicCount | | Integer | 板块的帖子数 |

10)用户查看所有的模块

|  |  |  |  |
| --- | --- | --- | --- |
| 用户查看所有的模块 | | | |
| 接口描述 | 用户查看所有的模块 | | |
| URL | /sector/allList | | |
| 请求类型 | get | | |
| 返回属性名 | | 数据类型 | 说明 |
| 1.err\_code | | Integer |  |
| 2.err\_msg | | String | 获取失败原因 |
| 3.data | | Object:list<SectorInfo> |  |
| 3.1.sectorId | | Integer | 板块编号 |
| 3.2.accountId | | String | 版主的帐号 |
| 3.3.sectorName | | String | 板块名字 |
| 3.4.clickingRate | | Integer | 板块的点击数 |
| 3.5.topicCount | | Integer | 板块的帖子数 |
| 3.6.accountName | | String | 版主名字 |

11)点击模块显示所对应的帖子

|  |  |  |  |
| --- | --- | --- | --- |
| 点击模块显示对应的帖子 | | | |
| 接口描述 | 点击模块显示对应的帖子 | | |
| URL | /topic/sectorList | | |
| 请求类型 | Post | | |
| 参数名 | | 数据类型 | 说明 |
| sectorId | | Integer | 板块编号 |
| 返回属性名 | | 数据类型 | 说明 |
| 1.err\_code | | Integer |  |
| 2.err\_msg | | String | 获取失败原因 |
| 3.data | | Object:list<TopicInfo> |  |
| 3.1.accountName | | String | 发帖人的名字 |
| 3.2.title | | String | 帖子的标题 |
| 3.3.topicText | | String | 帖子的正文 |
| 3.4.topicDate | | String | 发帖的时间 |
| 3.5.clickingRate | | Integer | 点击数 |
| 3.6.replyCount | | Integer | 帖子回复数 |

12)版主删除不合规的帖子

|  |  |  |  |
| --- | --- | --- | --- |
| 版主删除不合规的帖子 | | | |
| 接口描述 | 版主删除不合规的帖子 | | |
| URL | /topic/delect | | |
| 请求类型 | post | | |
| 参数名 | | 数据类型 | 说明 |
| topicId | | Integer | 帖子的编号 |
| 返回属性名 | | 数据类型 | 说明 |
| 1.err\_code | | Integer |  |
| 2.err\_msg | | String | 获取失败原因 |
| 3.data | | String | 返回1表示成功,0即为失败 |

13)管理员删除板块

|  |  |  |  |
| --- | --- | --- | --- |
| 管理员删除板块 | | | |
| 接口描述 | 管理员删除板块 | | |
| URL | /sector/delect | | |
| 请求类型 | Post | | |
| 参数名 | | 数据类型 | 说明 |
| sectorId | | Integer | 板块编号 |
| 返回属性名 | | 数据类型 | 说明 |
| 1.err\_code | | Integer |  |
| 2.err\_msg | | String | 获取失败原因 |
| 3.data | | String | 返回1表示成功,0即为失败 |

14)封号

|  |  |  |  |
| --- | --- | --- | --- |
| 封号 | | | |
| 接口描述 | 封号 | | |
| URL | /accouont/delete | | |
| 请求类型 | Post | | |
| 参数名 | | 数据类型 | 说明 |
| accountId | | String | 用户帐号 |
| 返回属性名 | | 数据类型 | 说明 |
| 1.err\_code | | Integer |  |
| 2.err\_msg | | String | 获取失败原因 |
| 3.data | | String | 返回1表示成功,0即为失败 |

15)解封帐号

|  |  |  |  |
| --- | --- | --- | --- |
| 解封帐号 | | | |
| 接口描述 | 解封帐号 | | |
| URL | /account/resume | | |
| 请求类型 | post | | |
| 参数名 | | 数据类型 | 说明 |
| accountId | | String | 用户帐号 |
| 返回属性名 | | 数据类型 | 说明 |
| 1.err\_code | | Integer |  |
| 2.err\_msg | | String | 获取失败原因 |
| 3.data | | String | 返回1表示成功,0即为失败 |

16)管理员查看所有的帐号

|  |  |  |  |
| --- | --- | --- | --- |
| 管理员查看所有的帐号 | | | |
| 接口描述 | 管理员查看所有的帐号 | | |
| URL | /account/list | | |
| 请求类型 | get | | |
| 返回属性名 | | 数据类型 | 说明 |
| 1.err\_code | | Integer |  |
| 2.err\_msg | | String | 获取失败原因 |
| 3.data | | Object:list<AccountInfo> |  |
| 3.1.accountId | | String | 帐号 |
| 3.2.accountName | | String | 帐号名字 |
| 3.3.iphoneNumber | | String | 电话号码 |
| 3.4.statusName | | String | 状态名字 |
| 3.5.accountStatus | | String | 用户状态编号 |

17)返回帖子数

|  |  |  |  |
| --- | --- | --- | --- |
| 返回帖子数 | | | |
| 接口描述 | 返回帖子数 | | |
| URL | /topic/number | | |
| 请求类型 | get | | |
| 返回属性名 | | 数据类型 | 说明 |
| 1.err\_code | | Integer |  |
| 2.err\_msg | | String | 获取失败原因 |
| 3.data | | Integer | 帖子数 |

18)帖子的模糊查询

|  |  |  |  |
| --- | --- | --- | --- |
| 帖子的模糊查询 | | | |
| 接口描述 | 帖子的模糊查询 | | |
| URL | /topic/check | | |
| 请求类型 | Post | | |
| 参数名 | | 数据类型 | 说明 |
| topicName | | String | 模糊查询的字眼 |
| 返回属性名 | | 数据类型 | 说明 |
| 1.err\_code | | Integer |  |
| 2.err\_msg | | String | 获取失败原因 |
| 3.data | | Object:list<TopicInfo> |  |
| 3.1.accountId | | String | 发帖人的帐号 |
| 3.2.accountName | | String | 发帖人的名字 |
| 3.3.topicId | | Integer | 帖子编号 |
| 3.4.topicDate | | String | 发帖的日期 |
| 3.5.clickingRate | | Integer | 帖子点击数 |
| 3.6.replyCount | | Integer | 帖子回复数 |
| 3.7.title | | String | 帖子标题 |
| 3.8.topicText | | String | 帖子正文 |
| 3.9.sectorId | | Integer | 板块的编号 |
| 3.10.sectorName | | Integer | 板块的名字 |

19)提升权限

|  |  |  |  |
| --- | --- | --- | --- |
| 提升权限 | | | |
| 接口描述 | 提升权限 | | |
| URL | /account/up | | |
| 请求类型 | post | | |
| 参数名 | | 数据类型 | 说明 |
| accountId | | String | 用户帐号 |
| sectorName | | String | 板块名字 |
| 返回属性名 | | 数据类型 | 说明 |
| 1.err\_code | | Integer |  |
| 2.err\_msg | | String | 获取失败原因 |
| 3.data | | String | 返回1表示成功,0即为失败 |

20)取消权限

|  |  |  |  |
| --- | --- | --- | --- |
| 取消权限 | | | |
| 接口描述 | 取消权限 | | |
| URL | /account/down | | |
| 请求类型 | post | | |
| 参数名 | | 数据类型 | 说明 |
| accountId | | String | 用户帐号 |
| 返回属性名 | | 数据类型 | 说明 |
| 1.err\_code | | Integer |  |
| 2.err\_msg | | String | 获取失败原因 |
| 3.data | | String | 返回1表示成功,0即为失败 |

21)创建板块

|  |  |  |  |
| --- | --- | --- | --- |
| 创建板块 | | | |
| 接口描述 | 创建板块 | | |
| URL | /sector/establish | | |
| 请求类型 | post | | |
| 参数名 | | 数据类型 | 说明 |
| sectorName | | String | 板块名字 |
| 返回属性名 | | 数据类型 | 说明 |
| 1.err\_code | | Integer |  |
| 2.err\_msg | | String | 获取失败原因 |
| 3.data | | String | 返回1表示成功,0即为失败 |

22)删除回复

|  |  |  |  |
| --- | --- | --- | --- |
| 删除回复 | | | |
| 接口描述 | 删除回复 | | |
| URL | /reply/delete | | |
| 请求类型 | post | | |
| 参数名 | | 数据类型 | 说明 |
| replyId | | Integer | 回复编号 |
| 返回属性名 | | 数据类型 | 说明 |
| 1.err\_code | | Integer |  |
| 2.err\_msg | | String | 获取失败原因 |
| 3.data | | String | 返回1表示成功,0即为失败 |

23)点赞回复

|  |  |  |  |
| --- | --- | --- | --- |
| 点赞回复 | | | |
| 接口描述 | 点赞回复 | | |
| URL | /reply/up | | |
| 请求类型 | post | | |
| 参数名 | | 数据类型 | 说明 |
| replyId | | Integer | 回复编号 |
| 返回属性名 | | 数据类型 | 说明 |
| 1.err\_code | | Integer |  |
| 2.err\_msg | | String | 获取失败原因 |
| 3.data | | String | 返回1表示成功,0即为失败 |

**5、组件设计**

**5.1、权限验证模块设计**

前端:获取帐号和密码以及用户选择的帐号类型,通过后端接口发送post请求方式进行帐号权限验证。

后端:根据前端传过来的数据对数据库进行配对,返回最终结果。

**5.2、功能模块详细设计**

增加板块:管理员设置板块的板块名,并且调用接口将数据写入数据库中。

删除板块:管理员选择不合规的板块,并通过接口post请求将对应的板块进行删除,然后相关联的帖子回复也会一起给删除。

授权管理:管理员可以通过用户列表,选择想提升权限的用户,选择想将其设为哪个板块的版主,通过对应ID和板块编号调用接口对数据库进行修改。

封号/解封操作:管理员可以通过用户列表,将不合规的用户进行封号,通过对应的帐号ID调用接口对数据库帐号状态进行修改。可以将封号的帐号进行解封,通过对应的帐号ID调用接口对数据库帐号状态进行修改。

修改密码/忘记密码:用户通过输入自己的帐号ID、帐号绑定的电话号码以及新密码进行修改密码,调用接口对数据库进行修改时,会先判断帐号ID与输入的电话号码是否相匹配,若相匹配就会将密码修改为新密码,若不匹配,则会返回修改失败。

删除帖子:管理员、所对应板块的版主以及发布该帖子的用户可以调用接口, 通过获取对应帖子的ID对数据库存放帖子的表进行删除操作。

发布帖子:用户可以通过输入帖子名字、帖子正文以及图片然后调用接口对数据库存放帖子的表进行插入操作。

查看自己的帖子:用户可以通过自己的帐号id调用接口查看自己所发布的所有帖子信息。

查看所有帖子:用户可以通过get请求方式获取道所有帖子的信息。

回复帖子:通过帖子的编号以及回复内容对数据库存放回复的表进行插入操作。

模糊查询:用户可以随意输入模糊字眼,然后通过对应接口对数据库中存放帖子的表进行like操作的模糊查询,若存在则会返回相关帖子的信息。

图文贴操作:用户在注册帐号、发布帖子以及回复帖子时,都可以进行发送图片的操作,系统将会将图片存放在云端上,然后调用接口将对应的路径发送给数据库进行插入修改操作。

**6、系统实现**

**6.1、关键代码分析**

前端:

(1)赋值全局变量：

this.setData({

user:name

})

(2)页面跳转：

wx.navigateTo({

url: '/pages/forget/forget',

})

},

(3)数据传送：

wx.setStorageSync('accountId', accountId);

(4)数据接收：

onLoad: function (options) {

var accountid = wx.getStorageSync('accountId');

this.setData({

accountId: accountid,

})

后端:使用的是SSM(Spring+SpringMVC+MyBatis)框架集由Spring、MyBatis两个开源框架整合而成,SpringMVC是Spring中的部分内容,最后接口返回统一的json数据, 接口包括的三大要素：状态码、结果集、成功或错误的提示,下面则是返回json数据的类

public class Result {

int err\_code;

String err\_msg;

Object data;

public Result(int err\_code, String err\_msg, Object data) {

this.data = data;

this.err\_code = err\_code;

this.err\_msg = err\_msg;

}

public int getErr\_code() {

return err\_code;

}

public void setErr\_code(int err\_code) {

this.err\_code = err\_code;

}

public String getErr\_msg() {

return err\_msg;

}

public void setErr\_msg(String err\_msg) {

this.err\_msg = err\_msg;

}

public Object getData() {

return data;

}

public void setData(Object data) {

this.data = data;

}

}

当调用某个接口时最终会收到返回结果,如图5。

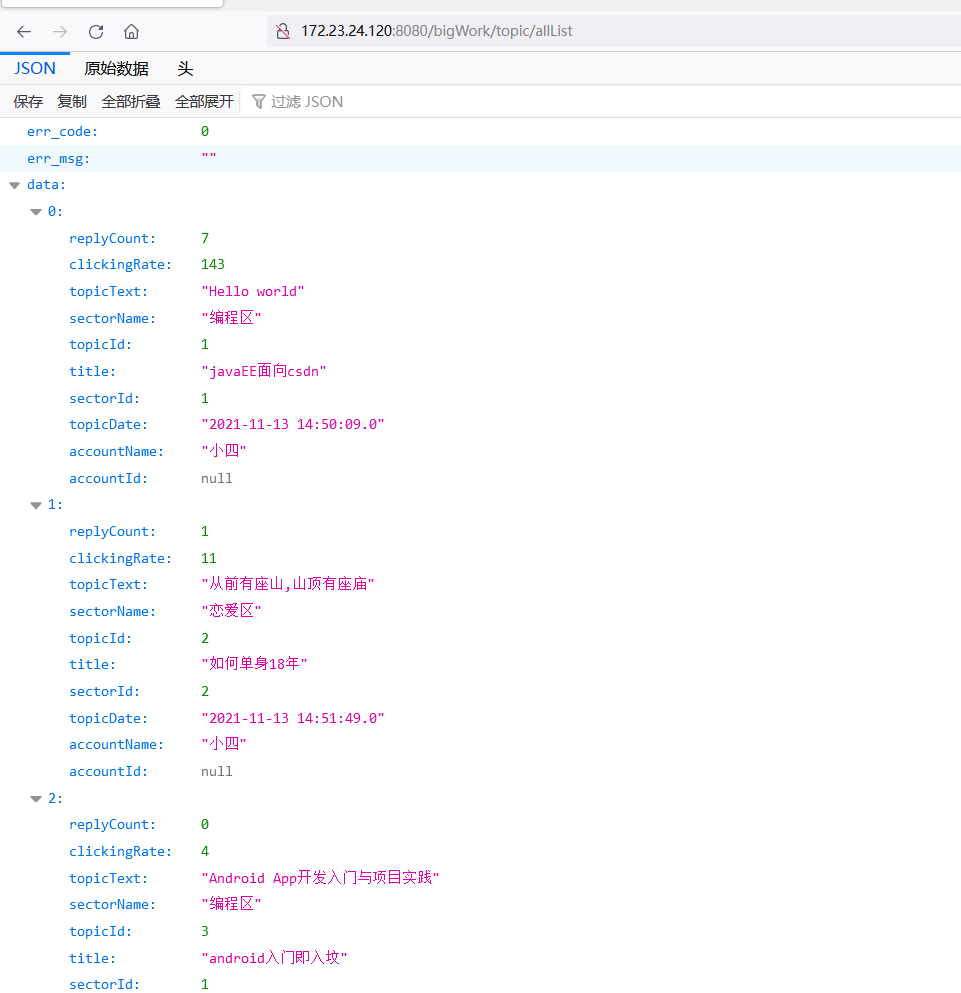


图5 json返回数据

**6.2、疑难问题解决**

前端: 遇到的问题是之前完全没解除过小程序的代码，所以在一开始学习的时候花费的许多问题，虽然和Html的代码很相似，比如div和view会经常搞混，后面上手了就变得简单一些，困难的点主要是界面路由跳转时数据传输的问题，这花费我大量的时间在官网看API，还有就是发图文贴的问题，用直接发生给数据库图片的地址后，再获取数据库的图片地址后显示不出来，所以我用了小程序自带的云开发来实现图文帖。还有就是社区界面，因为用了swiper导致有点静态，代码比较冗长。

后端:遇到的问题主要是搭建框架时的环境配置,以及配置文件的创建,遇到最多的问题就是版本不兼容,不能正常运行,最后只能慢慢调试版本,开始使用tomcat10的时候,与springweb有版本不兼容的问题,最后选择了9.0的版本。在写配置文件的时候,都是细小的细节导致代码没法正常运行,而且代码不报错,需要仔细地看。

**7、项目的发布和测试**

**7.1、项目发布过程**

项目的发布采用单机架构部署，后端项目部署在Java服务器Tomcat上，前端项目则是通过静态腾讯托管。

**7.2、项目测试过程**

|  |  |
| --- | --- |
| 管理员功能 | 是否实现 |
| 管理板块 | 已实现 |
| 管理帖子 | 已实现 |
| 管理版主 | 已实现 |
| 管理普通用户 | 已实现 |
| 对用户的封号和解封 | 已实现 |

表6 管理员功能测试过程表

|  |  |
| --- | --- |
| 版主功能 | 是否实现 |
| 管理帖子 | 已实现 |

表7 版主功能测试过程表

|  |  |
| --- | --- |
| 普通用户功能 | 是否实现 |
| 发帖 | 已实现 |
| 回帖 | 已实现 |
| 查看自己的帖子 | 已实现 |
| 查看所有的帖子 | 已实现 |
| 帖子的模糊搜索 | 已实现 |
| 图文贴 | 已实现 |

表8 普通用户功能测试过程表

**8、分析程序的优点和不足,遇到的困难及解决的问题,总结自己的收获**

优点:前端使用了微信小程序的框架实现了作业的基本功能，有较为简洁的界面，让人耳目一新,而后端则是使用了SSM框架的开发,使项目能够实现前后端分离,使整个项目的开发能够高效率的完成,不用互相等待,能够同时开发。

不足:前端功能路由跳转数据传输这方面写得差，导致一个页面有很多的变量。没有将所有的扩展功能全部实现,由于能力以及时间有限,上传下载文件和播放视频的功能没有完成,使得项目不够完美。

困难及解决的问题:

1、tomcat10.0版本与使用的spring版本不兼容,导致程序一直运行不了,最终选择9.0版本后能够正常运行。

2、第一次接触ssm框架的开发,对功能如何实现都一窍不通,后通过课本和网上资料,逐渐的了解到功能实现的步骤,先写实体类pojo，定义对象的属性，然后写Mapper.xml（Mybatis），其中定义你的功能，对应要对数据库进行的那些操作，比如 insert、selectAll、selectByKey、delete、update等,接着写对应的接口,紧接写Service.java，为控制层提供服务，接受控制层的参数，完成相应的功能，并返回给控制层,最后写Controller.java，连接页面请求和服务层，获取页面请求的参数，通过自动装配，映射不同的URL到相应的处理函数，并获取参数，对参数进行处理，之后传给服务层。

3、前端主要是上传图片的困难吧，之前直接用前端小程序和后端数据库对接的方式搞了很久都没成功，后面用云服务成功解决问题

收获: 让我懂得了合作的重要性，前后端开发人员一定要频繁沟通，沟通需要的数据以及怎么样的接口，让项目进行并行开发，而不是前端一定要等后端，这样效率不高，效率的最大化应该是前后端有效沟通并且同时开发。

**9、参考目录,写出自己编写该程序时阅读的书籍和出处**

参考书籍：

1、《Java EE框架整合开发入门到实战》 清华大学出版社

2、《零基础学Java》 吉林大学出版社

参考学习视频：

《尚硅谷2021版微信小程序开发（零基础小程序开发入门到精通）》

up主:尚硅谷 视频地址：尚硅谷2021版微信小程序开发（零基础小程序开发入门到精通）