

# BugShow 作业讨论平台

## 软件系统重构报告

【不咕鸟开发小组】



【同济大学 电子与信息工程学院】

2020-8-19

## 一、 软件重构任务简述

### 1. 重构动机

- a) 在之前的系统设计中对于某些功能没有进行角色上的区分，使用了相同的操作，而导致了一些 bug，这将会影响到作业讨论平台的正常使用。
  - b) 原先项目文件的划分存在不合理之处，导致有一些很具体的功能也被划分出单独的模块，不能很好地和其他模块形成同级并列关系。
  - c) 在之前的需求规格说明的基础上我们试图做出进一步的完善，从而增添了相应的功能，但最终未能实现而导致存在无用代码。
  - d) 原先项目代码存在一些不规范之处，且缺乏相应的注释，不便于理解。
- 基于以上原因，我们决定对于《软件工程》课程中的 BugShow 作业讨论平台项目进行软件重构。

### 2. 重构计划以及人员分工

#### 重构计划

时间	任务内容
7.23~8.04	前端：复习 bootstrap 和 JavaScript 的内容，对前端代码进行相关的解读，边解读边撰写相关注释以及分析现有代码的不合理之处。 后端：复习 flask 框架的相关使用，复习数据库模式，对后端代码进行相关的解读，边解读边撰写相关注释以及分析现有代码的不合理之处。
8.05~8.14	前端：对前端代码进行公共函数提取，并顺应后端的修改做出参数约定方面的变动。 后端：对后端代码进行架构上的重构，同时关注局部语句的细节错误和上下文的逻辑错误，修复相关 bug。
8.15~8.20	前后端人员共同完成测试报告与重构报告。

#### 人员分工

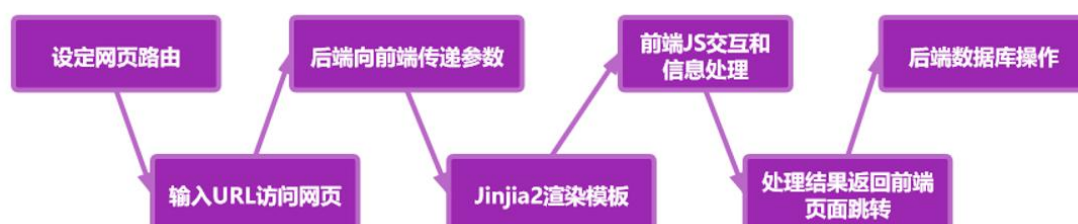
类型	人员	分工
前端	赵昊堃	对于前端公共函数进行抽取。
	陈博凡	去除前端无用代码，修改局部 bug。
	里雨妍	与后端进行衔接，修改相关的前后端参数约定。
后端	唐笠轩	对于后端软件架构进行调整，抽取相关的公共函数，详细分析了现有架构和代码的优势与不足之处。
	王天	去除后端无用代码，修改局部 bug，对于后端软件架构进行调整。
	欧阳君桦	与前端进行衔接，修改相关的前后端参数约定。

## 二、 现有软件系统分析

### 1. 软件系统架构分析

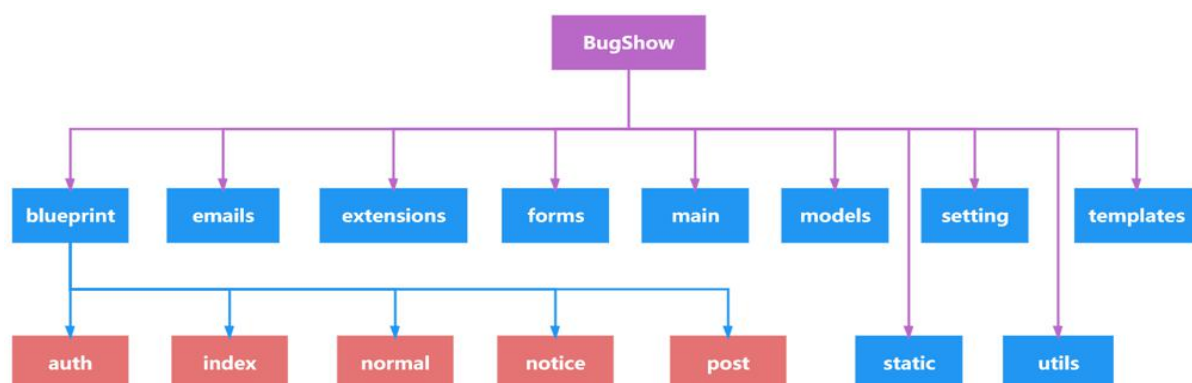
#### Flask 路由框架：

应用 Flask 框架进行开发，通过构造路由图来进行整个作业讨论平台结构的设计。



路由图使得后端的各个响应函数和前端的网址对应起来，在跳转到某网址时，将由与之对应的后端响应函数进行请求处理，将处理结果再返回给 WSGI 服务器，WSGI 服务器将接收到的参数（即响应函数的处理结果）传递给前端模板，由 Jinja2 引擎进行渲染从而完成一次完整的网页访问。

#### 项目总体构成：



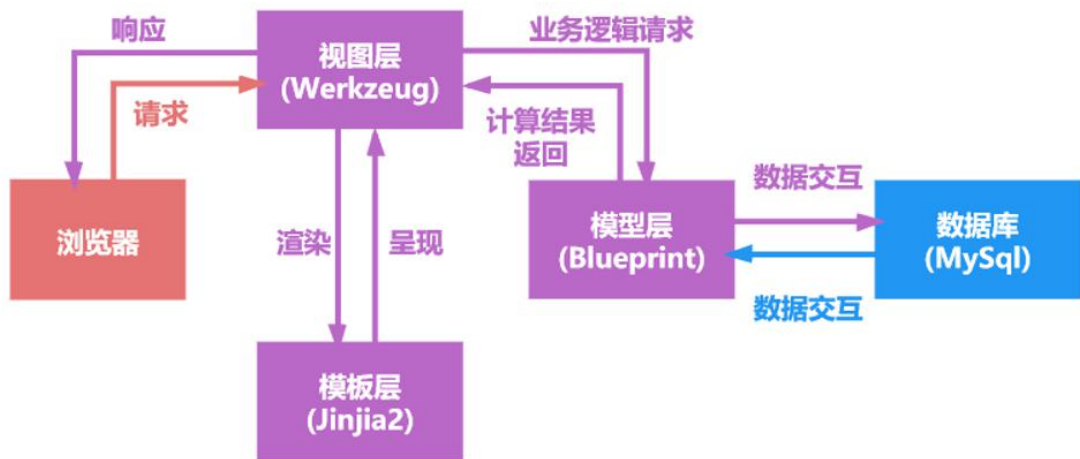
- **blueprint 文件夹：**工程蓝图，是整个业务模型的核心部分，与系统设计中的路由图 对应，实现了在不同网页位置处理请求、产生响应的各种功能函数。
- **emails：**包含了与邮件发送相关的功能函数。
- **extensions：**包含了对于系统设计中涉及到的功能组件进行导入和初始化相关的功能函数。
- **forms：**包含了在前后端交互过程中提交的表单类。
- **main：**包含了项目启动、运行的主函数。
- **model：**包含了对象关系映射所产生的虚拟数据库表对应的类。
- **settings：**包含了项目运行时相关环境变量的配置选项。
- **static 文件夹：**静态资源，用于存储项目前端所涉及到的样式资源。
- **templates 文件夹：**模板，用于存储项目前端的基本架构，运行时接受后端传递来的参数，由 Jinja2 引擎渲染，同时获取其他相关的静态资源，从而呈现出前端的显示和交互效果。
- **utils：**包含了与项目具体功能无关的常用工具函数。

**评价：**分析上述架构发现，虽然 utils 模块设置的初衷是为了方便其余模块调用共用的工具函数，这也是许多编程语言中对于工具库的常用手段，但在系统实现后发现该模块内工具函数并未得到充分的共用，且内部内聚程度极低，结构松散，从而在另一方面无疑增加了其他各个模块间的耦合程度，也不便于在对应模块中直接修改所需的对应功能。

2. 代码味道分析

a) 设计模式相关

MVT 架构示意图：



M 代指 Model，封装对数据库层的访问，通过面向对象编程的方式操作数据库。  
V 代指 View，接收 Http Request，处理业务逻辑并返回 Http Response。  
T 代指 Template，内嵌模板引擎，用于封装构造返回的 html，实现数据的动态展示。

**评价：**应用该设计模式，能够通过分层解耦的方式，将数据处理和界面显示的代码加以分离，使得程序能在不修改数据处理相关逻辑的情况下切换不同显示界面，从而提高程序的扩展性和可维护性。

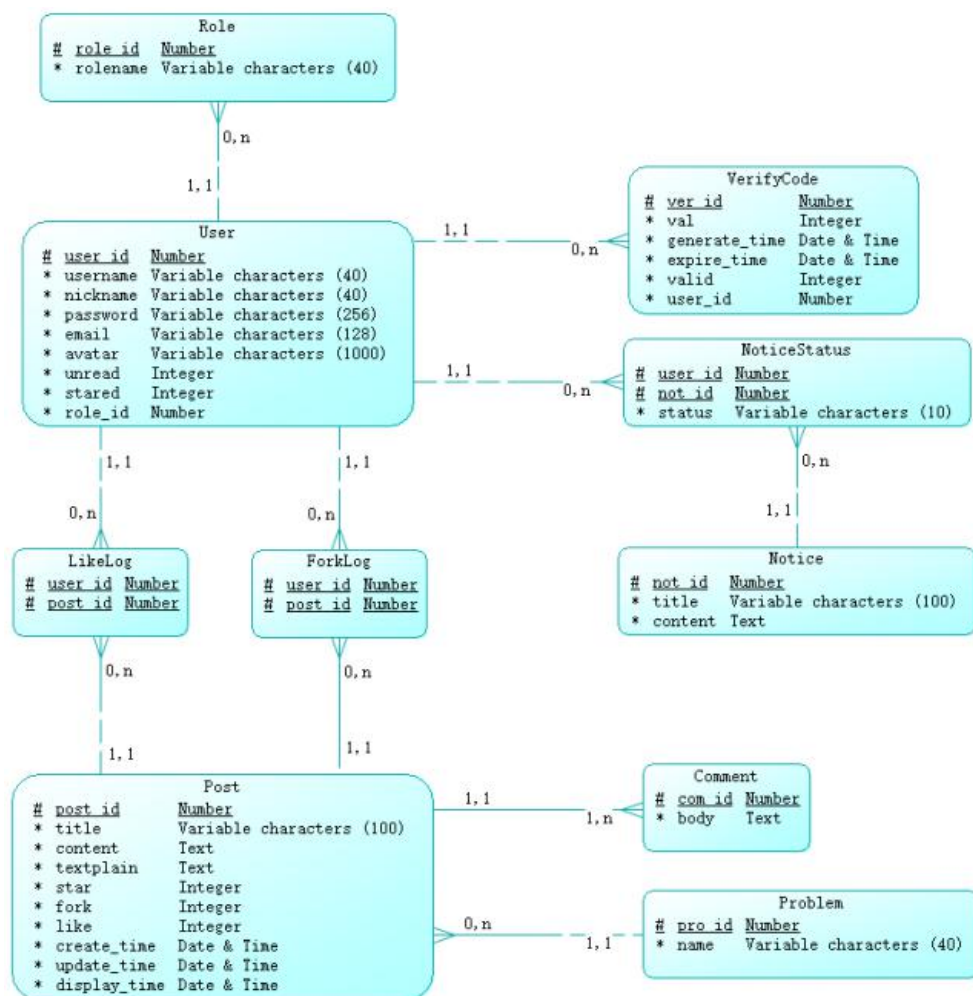
b) 类封装相关

models 虚拟数据库表对应类	所包含字段
用户信息表	用户 id, 用户账号, 用户昵称, 用户密码, 电子邮箱, 头像 URL, 未读通知数量、收藏资料数量
角色信息表	角色 id, 角色名称
问题信息表	问题 id, 问题名称
讨论信息表	讨论 id, 讨论标题, 讨论内容格式, 讨论内容文本, 解答标记, 点赞次数, 收藏次数, 创建时间, 更新时间, 开放显示时间, 问题 id, 用户 id
评论信息表	评论 id, 评论内容, 讨论 id, 用户 id
验证码信息表	验证码 id, 验证码内容, 产生时间, 过期时间, 有效性, 用户 id
通知信息表	通知 id, 通知标题, 通知内容, 用户 id
通知状态表	用户 id, 通知 id, 通知状态
点赞表	用户 id, 资料 id

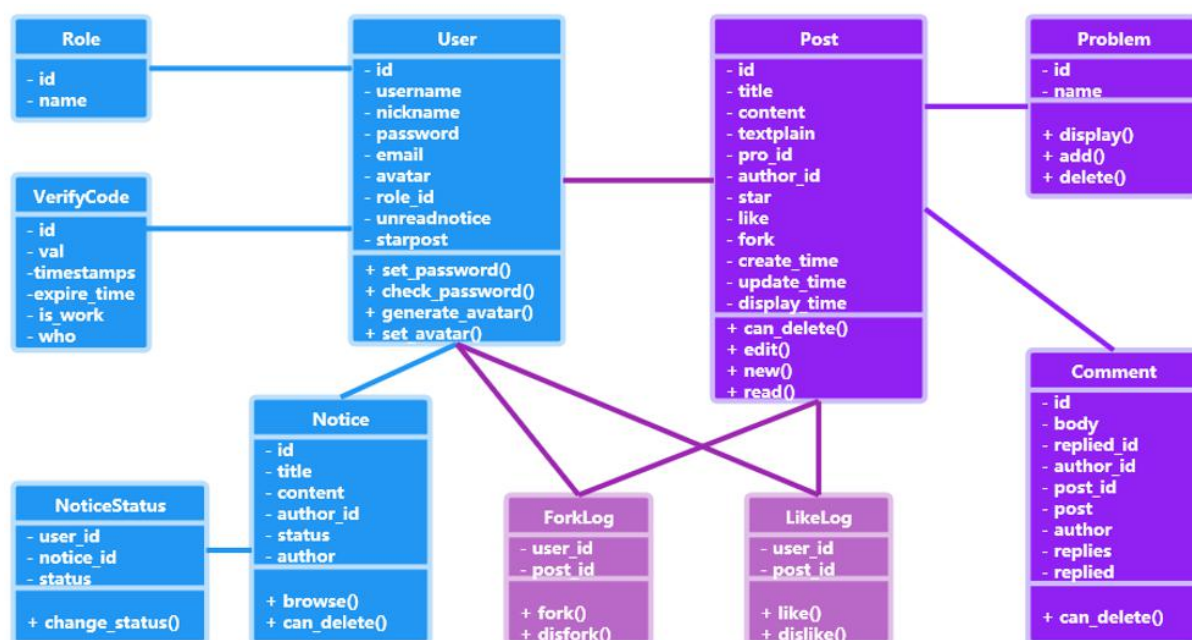
收藏表

用户 id, 资料 id

虚拟数据库表对应类的实体关系图



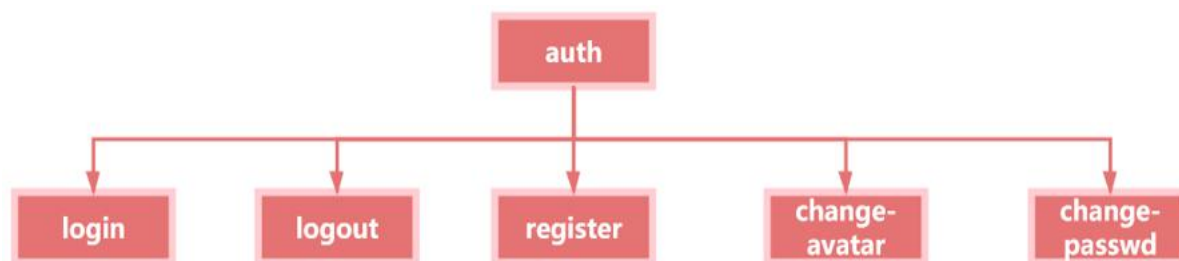
对应的类设计



评价：数据库类设计部分在当前应用背景下较为合理。

### c) 方法抽取相关

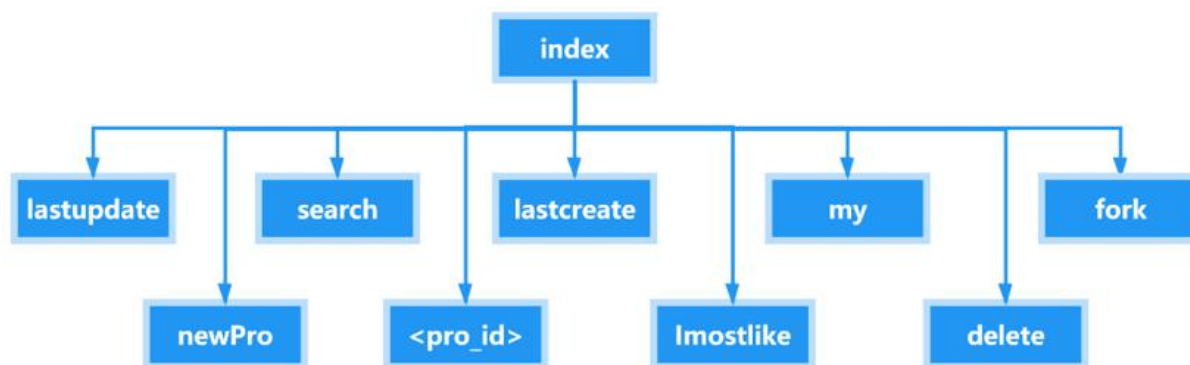
1. **auth 模块**：负责用户账户个人信息的注册登记与管理等。



抽取的方法：

- **register()**：通过发送验证码到指定邮箱的方式注册用户账号。
- **login()**：通过邮箱/用户名及密码的方式登录平台。
- **ChangePassword()**：支持用户修改密码。
- **ChangeAvatar()**：支持用户修改头像。
- **logout()**：用户登出平台。

2. **index 模块**：负责页面跳转和整体效果的显示相关。



抽取的方法：

- **newPro()**：添加新题目。
- **indexpro\_lastupdate(pro\_id)**：显示最近的更新。
- **indexpro\_lastcreate(pro\_id)**：显示最近的发布。
- **indexpro\_mostlike(pro\_id)**：显示感兴趣的题目。
- **deletePro(pro\_id)**：删除题目。
- **indexpro\_my(pro\_id)**：显示当前用户发布的题目。
- **indexpro\_fork(pro\_id)**：显示当前用户的收藏。
- **indexpro\_search()**：显示搜索到的题目。



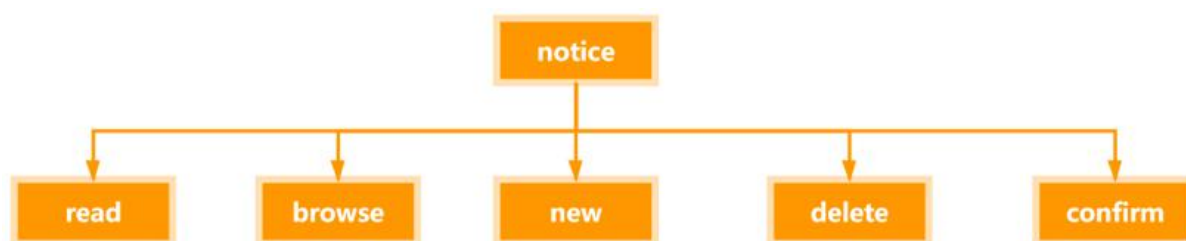
3. **normal 模块**：包含常用的工具业务功能。

抽取的方法：

- **send()**：根据用户名、邮箱账号，发送验证码到邮箱。
- **image\_upload()**：上传图片。

**评价**：分析得知，在原本的设计中，由于通过电子邮件发送验证码的功能在实现时具有一定的难度，因此安排了专门的人员进行编写，从而导致 email 功能单独形成了模块，代码编写在单独的文件中。该部分应当集成在 normal 模块中，通过以蓝图中一些通用功能函数路由的方式加以集中。

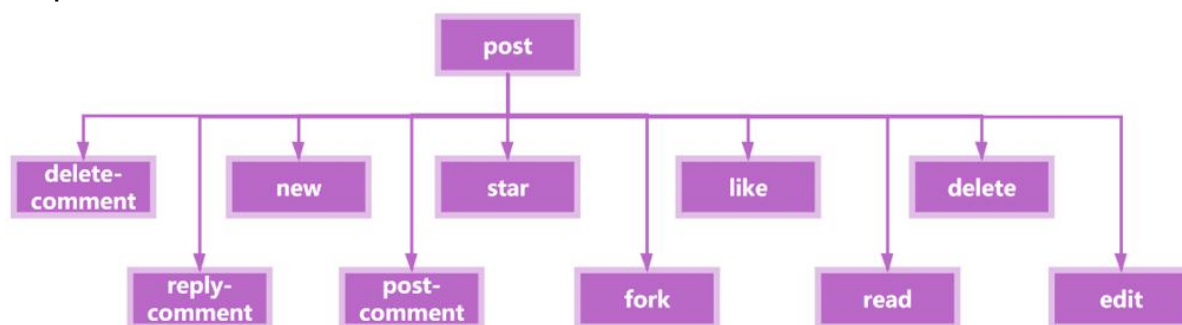
4. **notice 模块**：负责通知功能的管理。



抽取的方法：

- **new\_notice()**：发布新的通知。
- **browse\_notice()**：显示所有的通知项。
- **delete(notice\_id)**：删除通知。
- **confirm(notice\_id)**：通知的确认。
- **read(notice\_id)**：阅读指定通知。

5. **post 模块**：负责平台资料的管理。



抽取的方法：

- **new\_post()**：发布新的帖子。
- **read(post\_id)**：阅读指定帖子。
- **edit(post\_id)**：编辑帖子。
- **delete(post\_id)**：删除帖子。
- **post\_comment()**：评论帖子。
- **reply\_comment()**：回复帖子。
- **star(post\_id)**：对解答进行标星或取消标星。
- **like(post\_id)**：对解答进行点赞或取消点赞。

- **fork(post\_id):** 对帖子进行收藏或取消收藏。
- **delete\_comment(comment\_id):** 删除评论。

**该部分总体评价:** 在现有的项目代码中，路由函数是需要进行一系列数据库操作从而实现相应的业务逻辑，原软件系统的 auth、index、post 等文件中的路由函数较为臃肿，结构不清晰，存在着代码复制粘贴造成的冗余现象，不易于阅读和理解，使得项目文件冗余且不利于维护。因此，应当将蓝图中相关模块的功能语句和模型中类的方法进行整合，提取出公共函数，在蓝图的路由函数中直接调用模型的方法实现与以往相同的业务逻辑。

d) 变量相关

当前的软件中变量命名存在以下两方面问题:

- 由于分工时组员未统一标准，不同模块间出现了驼峰命名法和下划线命名法混用的情况:

```
73 @auth_bp.route('/change-passwd', methods=['GET', 'POST'])
74 def ChangePassword():
13 @notice_bp.route('/new', methods=['GET', 'POST'])
14 @login_required
15 def new_notice():
```

后续决定统一为下划线命名法。

- 个别变量的命名表述不准确或不易理解

```
@index_bp.route('/index/search', methods=['POST', 'GET'])
@login_required
def indexpro_search():
```

e) Bug

存在的 Bug	重要性等级
学生删除自己收件箱中的通知后，其他学生查看相同通知会报错。	较高
用户头像图片无法显示。	一般
在问题列表中仅能存放有限数量的问题。	较高
页面跳转异常。	较高



### 三、 对软件系统的改进

#### 1. 对系统架构的改进（可选）

##### 改进 1

改进内容	将相关性较大的 <b>normal</b> 模块和 <b>email</b> 模块进行合并，使得响应代码集中在一个代码文件之中
改进原因	在原本的设计中，由于通过电子邮件发送验证码的功能在实现时具有一定的难度，因此安排了专门的人员进行编写，从而导致 <b>email</b> 功能单独形成了模块，代码编写在单独的文件中。 <b>normal</b> 模块则是蓝图中一些通用功能函数路由的集中。而实际上最终 <b>normal</b> 模块仅包含发送电子邮件功能函数的路由。
改进结果	将 <b>email</b> 模块合并到 <b>normal</b> 模块中，从而避免了单独的 <b>email</b> 功能和配置文件、模型文件处于并列层次的不合理局面，同时增强了电子邮件功能对应子系统的内聚程度。
测试结果	通过

##### 改进 2

改进内容	将常用的功能函数直接放置在对应的需要使用其的模块中，而不是集中起来构成 <b>utils</b> 文件
改进原因	在原本的设计中，我们将常用的功能函数如生成验证码、判断是否为安全 URL、获取 md5 码、获取纯文本格式等统一放置在 <b>utils</b> 文件中，这也是许多编程语言中对于工具库的常用手段，但是这样会增加模块之间的耦合程度，也不便于在对应模块中直接修改所需的对应功能。
改进结果	<b>get_ver_code</b> 、 <b>get_md5</b> 等功能函数被分别改写到对应的 <b>normal</b> 模块位置，对一些未使用到的功能函数进行了删除，降低了模块间的耦合程度。
测试结果	通过

##### 改进 3

改进内容	将蓝图中相关模块的功能语句和模型中类的方法进行了整合，提取出公共函数，在蓝图的路由函数中直接调用模型的方法实现与以往相同的业务逻辑
改进原因	在现有的项目代码中，路由函数往往是需要进行一系列数据库操作从而实现相应的业务逻辑，这些代码是松散地呈列在路由函数的函数体中，存在着类似路由函数中需要将相同代码复制粘贴多遍的情况，使得项目文件冗余且不利于维护。
改进结果	将路由函数中的业务功能统一提取成公共函数，作为模型的方法，路由函数所需做的就是调用模型中的方法。提高了代码复用性，使得项目结构更加清晰，便于维护和阅读。
测试结果	通过

##### 改进 4

改进内容	将对通知的删除拆解成了学生和教师/管理员两部分
改进原因	原本的通知删除操作，对应三种角色只有一个路由函数，这样不仅导致了 bug，而且从架构角度也使得该路由函数显得冗余，未能按相应的功能进行精确划分。
改进结果	将原先一个路由函数拆解成两个，分别实现角色为学生时删除收件箱中的通知（在数据库中仅仅修改该用户和通知状态为“delete”）和角色为教师/管理员时删除已发布的通知（在数据库中真正删除对应的通知记录）两种业务功能。
测试结果	通过

## 2. 对设计模式的改进（可选）

### 改进 1

改进内容	变量与函数命名规范
改进原因	原有的部分变量与函数的命名不易理解，且存在驼峰命名法和下划线命名法混用的情况，在阅读代码时造成一定的障碍。
改进结果	变量与函数的命名都做到了见名知义，并且将混用驼峰命名法的部分统一修改为了下划线命名法，提高了代码的可读性，使代码阅读者一目了然，更容易理解代码逻辑。
测试结果	通过

### 改进 2

改进内容	函数方法抽取
改进原因	原软件系统的 auth、index、post 等文件中的路由函数较为臃肿，结构不清晰，存在着代码复制粘贴造成的冗余现象，不易于阅读和理解。
改进结果	将路由函数中的操作合理抽取为了模型中的方法，使路由函数变得简洁。灵巧，结构更为清晰，提高了代码的可读性，方便了代码重用。
测试结果	通过

### 改进 3

改进内容	完善注释
改进原因	原项目文件中缺少相关的注释，导致代码不便于理解，维护起来也比较困难。
改进结果	对于不易理解的、关键部分的代码添加了相关的注释，有效地提高了代码的可读性。
测试结果	通过

### 改进 4

改进内容	去除无用代码
改进原因	原项目文件中存在初期计划实现，而后由于技术和时间等因素没有实现的，超出需求规格说明的功能，这些功能编写了部分代码，然而较为残缺，并没有实际使用，对理解整体项目构成造成了一定困难，且使得代码文件存在冗余。
改进结果	对于实际并未使用到的变量、函数等进行了去除，从而使得整个项目各个文件得到了精简，且实际留存的都是真正有用的代码，提高了项目可读性和维护的容易程度。
测试结果	通过

### 3. 对 bug 的改进

#### 改进 1

改进内容	修复学生删除自己收件箱中的通知后，其他学生查看相同通知报错
改进原因	在原有代码中，对于通知的删除，不同角色使用了同一个路由函数，这使得学生删除自己收件箱中的通知时实际上将其记录从数据库中也删除了。
改进结果	将原先一个路由函数拆解成两个，分别实现角色为学生时删除收件箱中的通知（在数据库中仅仅修改该用户和通知状态为“delete”）和角色为教师/管理员时删除已发布的通知（在数据库中真正删除对应的通知记录）两种业务功能。
测试结果	通过

#### 改进 2

改进内容	修复用户头像图片无法显示
改进原因	用户上传到服务器上的头像图片，在用户主页以及回帖页面等处无法正常显示，但服务器中确实已经存储了对应的图片文件。
改进结果	将前端用于定位头像文件的字符串中所使用的“\\”修改为“/”以适应 CentOS 7 系统的路径表示形式，从而能够正确定位到用户上传的头像文件，能够正常显示头像。
测试结果	通过

#### 改进 3

改进内容	修复在问题列表中可以存放的问题数量有限
改进原因	作业个数超过 6 个之后，作业列表的背景边框没有跟着延申，作业列表会超出边界，显示效果需要优化。
改进结果	在前端对问题列表指定弹性盒子属性，从而使得问题列表能够随着问题数量的增减自动伸缩，保证问题既不超出问题列表，也不留下大量空白。
测试结果	通过

#### 改进 4

改进内容	修复页面跳转异常
------	----------

改进原因	在页面左上方存在“主页\讨论\当前讨论”的导航栏，点击主页返回的是讨论页面，点击讨论无法返回。
改进结果	在前端代码中，对导航栏的 href 设置一开始存在错误，有的设置为了“#”因此无法实现跳转，将其设置正确后，点击主页能够返回网站首页，点击讨论能够返回到具体的讨论条目页面。
测试结果	通过

## 四、 重构后软件系统测试报告

### 测试对象

对同济大学计算机科学与技术系不咕鸟小组开发的 BugShow 作业交流讨论平台进行测试。

### 测试人员

姓名	学号	测试时间
唐笠轩	1752367	8.15~8.18
王天	1853971	8.15~8.18
赵昊堃	1852877	8.15~8.18
欧阳君桦	1851451	8.15~8.18
陈博凡	1852661	8.15~8.18
里雨妍	1853223	8.15~8.18

### 测试环境

#### 硬件环境

“机型（配置）”：

操作系统	Windows 10
机型	Dell Inspiron 15-7510, Dell G7-7588
CPU	Intel(R) Core(TM) i7-11800H CPU @ 2.50GHz
硬盘大小	>1GB
内存	>1GB
带宽	>1MB

“用途及特殊说明”：此设备用于作为客户机供浏览器运行，通过 TCP/IP 协议和服务器进行交互，无其他特殊约束。

## 软件环境

“软件及版本”：Google Chrome 91.0.4472.114 及以上版本。

## 测试用例设计

测试用例设计原则：等价类划分、边界值、决策表、场景法等。

表格 1

功能点/issue	测试用例 id	测试用例	优先级
上传图片	1	能否正常上传图片	最高
上传图片	2	上传图片失败是否给出提示	高
上传图片	3	上传的图片格式合法	高
上传图片	4	上传的图片大小过大	高
上传图片	5	上传的图片格式后缀非法	高
上传图片	6	上传的图片名称长度是否有限制	高
上传图片	7	当上传的图片名称有特殊字符或中英文文字时能否正常上传	高
发表帖子	8	能否正常发表帖子	最高
发表帖子	9	设置同名帖子是否给出提示	高
发表帖子	10	能否删除帖子	最高
发表帖子	11	能否选择帖子所属的问题	最高
发表帖子	12	能否发表包含图片的帖子	最高
发表帖子	13	能否设置帖子标题	最高
发表帖子	14	帖子标题包含特殊字符或中英文文字时能否正常发表	最高
设置显示时间	15	能否在发表帖子时设置显示时间	最高
设置显示时间	16	能否更改已有帖子的显示时间	最高
删除问题	17	能否正常删除已有问题	最高
删除问题	18	删除问题时能否删除其下所有帖子	最高
发表通知	19	管理员/教师能否正常发表通知	最高
发表通知	20	管理员/教师能否设置通知标题	最高
发表通知	21	管理员/教师发表通知是否存在字数限制	高
删除通知	22	管理员/教师能否删除已经发布的通知	高
删除通知	23	学生能否删除自己收件箱中的通知	高
查看通知	24	学生能否查看自己收件箱中的通知并减少已读	最高
收藏帖子	25	能否收藏帖子	高

收藏帖子	26	能否取消已收藏帖子	高
标星帖子	27	教师能否将帖子设置为标准答案	高
标星帖子	28	教师能否取消标准答案帖子	高
查看用户	29	能否进入到其他用户的个人页面	最高
查看用户	30	能否关注其他用户	高
查看用户	31	能否对已经关注的作者取消关注	高
发表评论	32	能否对正在查看的作品发表评论内容	最高
置顶评论	33	能否对自己作品下的评论进行置顶操作	高
置顶评论	34	能否对自己作品下的置顶评论取消置顶	高
点赞评论	35	能否对作品下的评论内容进行点赞操作	高
点赞评论	36	被点赞的评论的发表者能否收到点赞通知	中
点赞评论	37	能否对自己已经点赞的评论取消点赞	高
举报评论	38	能否对作品下的评论内容进行举报操作	高
举报评论	39	被举报的评论的发表者能否收到举报通知	中
举报评论	40	管理员能否对被举报的评论进行审核	高
删除评论	41	能否删除自己作品下的评论内容	最高
删除评论	42	管理员能否删除被举报的评论内容	高
注册	43	用户名、密码、确认密码、邮箱合法	高
注册	44	用户名长度非法	高
注册	45	密码长度非法	高
注册	46	邮箱格式非法	高
注册	47	确认密码和密码不一致	高
注册	48	用户名或邮箱已被注册	高
注册	49	用户名包含非法脚本代码	高
登录	50	用户名和密码合法	高
登录	51	用户名错误	高
登录	52	密码错误	高
登出	53	账号退出	高
修改密码	54	原密码、新密码、确认密码合法	高
修改密码	55	原密码错误	高
修改密码	56	新密码不合法	高
修改密码	57	两次密码不一致	高
修改密码	58	用户未登录	高
修改用户名	59	新用户名合法且未被注册	高
修改用户名	60	新用户名不合法	高
修改用户名	61	新用户名已被注册	高
用户权限变更	62	管理员能够封禁已经激活的账号	中
用户权限变更	63	管理雨能够激活已经封禁的账号	中

优先级参考：

最高优先级（冒烟测试）：系统或模块是否可运行；



高优先级：功能点是否运行正确，稳定；  
中优先级：功能点是否处理了不影响正常使用的异常情况；  
低优先级：界面是否美观，性能是否满足要求；

# 测试结果与分析

## 需求覆盖分析

表格 2

测试用例 id	是否通过	备注
1	是	
2	是	
3	是	
4	是	
5	是	
6	是	
7	是	
8	是	
9	是	
10	是	
11	是	
12	是	
13	是	
14	是	
15	是	
16	是	
17	是	
18	是	
19	是	
20	是	
21	是	
22	是	
23	是	
24	是	
25	是	
26	是	
27	是	
28	是	
29	是	
30	是	
31	是	

32	是	
33	是	
34	是	
35	是	
36	是	
37	是	
38	是	
39	是	
40	是	
41	是	
42	是	
43	是	
44	是	
45	是	
46	是	
47	是	
48	是	
49	是	
50	是	
51	是	
52	是	
53	是	
54	是	
55	是	
56	是	
57	是	
58	是	
59	是	
60	是	
61	是	
62	是	
63	是	

## 测试覆盖分析

表格 3

功能点/issue	测试用例数	执行数	通过数
上传图片	7	7	7
发表帖子	3	3	3
设置显示时间	2	2	2
设置标题	5	5	5
删除帖子	2	2	2

查看帖子	10	10	10
查看用户	3	3	3
发表评论	1	1	1
置顶评论	2	2	2
点赞评论	3	3	3
举报评论	3	3	3
删除评论	2	2	2
发表通知	4	4	4
删除通知	2	2	2
查看通知	2	2	2
注册	7	7	7
登录	3	3	3
登出	1	1	1
修改密码	5	5	5
修改用户名	3	3	3
用户权限变更	2	2	2

缺陷分析

表格 4

测试内容	测试次数	是否出现问题	问题描述	问题严重程度	是否修复
注册	10	是	注册邮箱已被使用时，还能够进行验证码发送，在最后点击注册时才会提醒无法使用。 理论上在点击发送验证码时就应该检测邮箱是否能使用	高	否
登录	5	是	注册失败后手动进入登录页面，登录按钮点击后无反应。	中	否
修改密码	3	否			
修改头像	2	否			
查看作业分类下的讨论	5	是	对于没有讨论的作业分类应该给出一个简单的提示“当前作业下暂无讨论”	低	是
点击讨论查看内容	10	是	有图片无法显示的情况,包括评论中的用户的头像	中	是

评论	6	是	表情栏为空，帮助键无法点击，图片无法选择本地图片	中	否
讨论界面	5	是	在左上方存在“主页\讨论\当前讨论”的导航栏，点击主页返回的是讨论页面，点击讨论无法返回	中	是
标星功能	5	是	标星功能使用正常，但是可以对别人标星的讨论取消标记很不合理，并且建议不能对自己的讨论标星（这里当时需求没有写，但建议加上）	高	是
举报	5	是	讨论的举报按键点击后无反应	高	否
删除评论	3	否			
删除讨论	2	是	管理员账号都无法删除主题讨论，仍然显示的是举报	高	是
搜索	5	是	搜索问题后页面出现 405 错误	非常严重	是
发布讨论	3	是	添加图片，从本地上传时，网络错误，无法正确上传	高	
发布重复标题讨论	3	否			
删除讨论	1	否			
添加作业列表	5	是	在作业个数超过 6 个之后，作业列表的背景边框没有跟着延伸，作业列表会超出边界，显示效果需要优化	低	是
删除作业列表	3	否			
后台管理	3	是	学生用户在工作台中也能看见后台管理的选项，点击却	非常严重	否

			不能访问。 反而是管理员用户看不见后台管理。 现在完全无法进行后台管理		
发布通知	2	否			
接收通知	2	是	能正常查看，但是点击确认收到, 会 Internal Server Error	高	是

性能分析

测试环境：i7 11800H

测试工具：Locust

主要测试场景：

- ①用户同时登录注册
- ②用户同时上传图片

负载测试：

- ①在 100 个用户同时登录注册时，无错误，平均响应时间在 1s 以内，基本满足用户体验要求。
- ②在 10 个用户同时发表帖子时，无错误，响应时间在 10s 以内，略微影响用户使用体验。

压力测试：

- ①在 500 个用户同时登录注册时，错误率在 20%左右
- ②在 50 个用户同时发表帖子时，错误率在 44%左右

测试结论

软件 bug 数量较少，严重程度较低。版本功能基本实现且运行稳定，问题修改及时，在预定日期内完成开发和测试进度。系统测试通过，可以发布及上线。

五、 成员贡献度

姓名	贡献度
唐笠轩	100
赵昊堃	95
陈博凡	90
里雨妍	90
欧阳君桦	90
王天	100