Python课程设计报告

—— 指导教师：熊建国

赵鹏飞 软件1801班 12018111011

**一、需求分析**

**1.1.题目描述**

基于flask框架开发设计一个留言板系统。

用户登陆后能发布留言，能对自己发布的留言进行管理，能查看分页显示留言列表。

能根据留言的发布者，和留言内容，做模糊查询。

**1.2.功能设计**

1.2.1 登录与注册

* 1. 用户通过用户名、昵称、密码进行注册
  2. 用户通过用户名、密码进行登录
  3. 用户登录状态的保持

1.2.2 留言发布

1. 留言标题
2. 留言内容（富文本）
3. 留言分类
4. 保存留言草稿

1.2.3 留言阅读与评论

1. 留言评论（回复评论）
2. 留言阅读量
3. 留言回复量

1.2.4 个人信息管理

1. 显示用户注册时长
2. 上传头像
3. 修改昵称
4. 修改密码

1.2.5 个人留言管理

1. 留言修改
2. 留言隐藏
3. 评论管理
4. 我的消息（对我留言的评论，对我评论的回复）

1.2.6 全局搜索

* 1. 模糊搜索所有可见留言
  2. 模糊搜索用户

**二、详细设计**

**2.1.数据库设计**

表2.1.1 user用户表（标蓝约束非数据库约束，由后端程序控制）

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 字段约束 | 字段说明 |
| user\_id | INT | 主键、自增 | 用户唯一id |
| username | VARCHAR(50) | 非空、唯一 | 用户登录时输入账号 |
| nickname | VARCHAR(50) | 非空、唯一 | 用户昵称 |
| password | CHAR(32) | MD5加密字符 | 用户登录密码 |
| avatar | VARCHAR(50) | 非空，有默认头像 | 头像文件名称 |
| email | VARCHAR(50) | 可为空、唯一 | 用户安全邮箱 |
| role | VARCHAR(10) | 非空、默认值为user  admin为管理员  user为普通用户 | 用户身份 |
| create\_time | DATETIME | 创建时默认当前时间，格式**YYYY-MM-DD hh:mm:ss** | 记录创建时间 |
| update\_time | DATETIME | 创建和更新时默认当前时间，格式同上 | 记录修改时间 |

表2.1.2 message留言表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 字段约束 | 字段说明 |
| message\_id | INT | 主键、自增 | 留言唯一id |
| user\_id | INT | 外键、非空 | 发布留言的用户id |
| type | SMALLINT | 无默认值、非空 | 留言类型 |
| headline | VARCHAR(100) | 无默认值、非空 | 留言标题 |
| content | MEDIUMTEXT | 最长存储16777216个字符 | 留言内容 |
| read\_count | INT | 非空、默认为0 | 阅读次数 |
| reply\_count | INT | 非空、默认为0 | 回复数量（包括留言评论和评论的回复） |
| hidden | BIT(1) | 非空、默认为0 | 是否隐藏（临时删除） |
| drafted | BIT(1) | 无默认值、非空 | 是否为草稿 |
| recommended | BIT(1) | 非空、默认为0 | 管理员是否置顶 |
| create\_time | DATETIME | 创建时默认当前时间，格式**YYYY-MM-DD hh:mm:ss** | 记录创建时间 |
| update\_time | DATETIME | 创建和更新时默认当前时间，格式同上 | 记录修改时间 |

表2.1.3 comment评论表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 字段约束 | 字段说明 |
| comment\_id | INT | 主键、自增 | 评论唯一id |
| user\_id | INT | 外键、非空 | 发表回复的用户id |
| message\_id | INT | 外键、非空 | 回复的留言id |
| content | TEXT | 最多65536个字符 | 评论内容 |
| reply\_to | INT | 留言的原始评论为0  评论的回复该字段为对应的commentid | 回复针对的内容 |
| reply\_to\_id | INT | 外键、非空 | 回复针对的用户id |
| hidden | BIT(1) | 非空、默认为0 | 是否隐藏 |
| create\_time | DATETIME | 创建时默认当前时间，格式**YYYY-MM-DD hh:mm:ss** | 记录创建时间 |
| update\_time | DATETIME | 创建和更新时默认当前时间，格式同上 | 记录修改时间 |

**2.2 使用的框架和库**

2.2.1 前端：

Bootstrap -- 用于为组件提供美观的外观

Bootbox -- Bootstrap插件，用于显示美观的对话框

JQuery -- 用于简化JavaScript操作

MD5 -- 用于对密码传输进行加密

UEditor -- 百度富文本编辑器，用于对留言内容的编辑

2.2.2 后端

Flask -- Python Web框架

SQLAlchemy -- 数据库ORM

Jinja2 -- 模板引擎，用于后端渲染页面

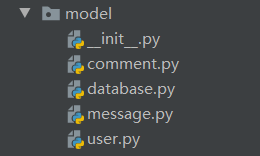
mysql-connector-python -- MySQL数据库驱动

**2.3 程序结构**

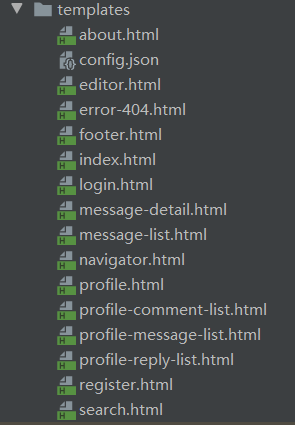
程序采用MVC架构，以业务逻辑、数据、显示界面分离的方式组织代码，保证了代码的可读性和可扩展性。

Model使用SQLAlchemy映射到数据库，将用户数据持久化到数据库，并提供数据存取的接口； 前端页面View由Jinja2渲染，将数据显示给用户，并接受用户的输入；Controller负责Model和View间的同步，当用户发出请求时，Controller从Model中读取数据更新View，用户输入数据时，Controller检查数据的合理性并将数据更新到Model。

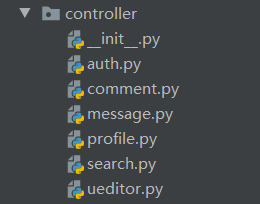
最终项目结构如下：



**图2.3.1 Model按映射的数据库表进行分类**



**图2.3.2 View按页面进行分类**

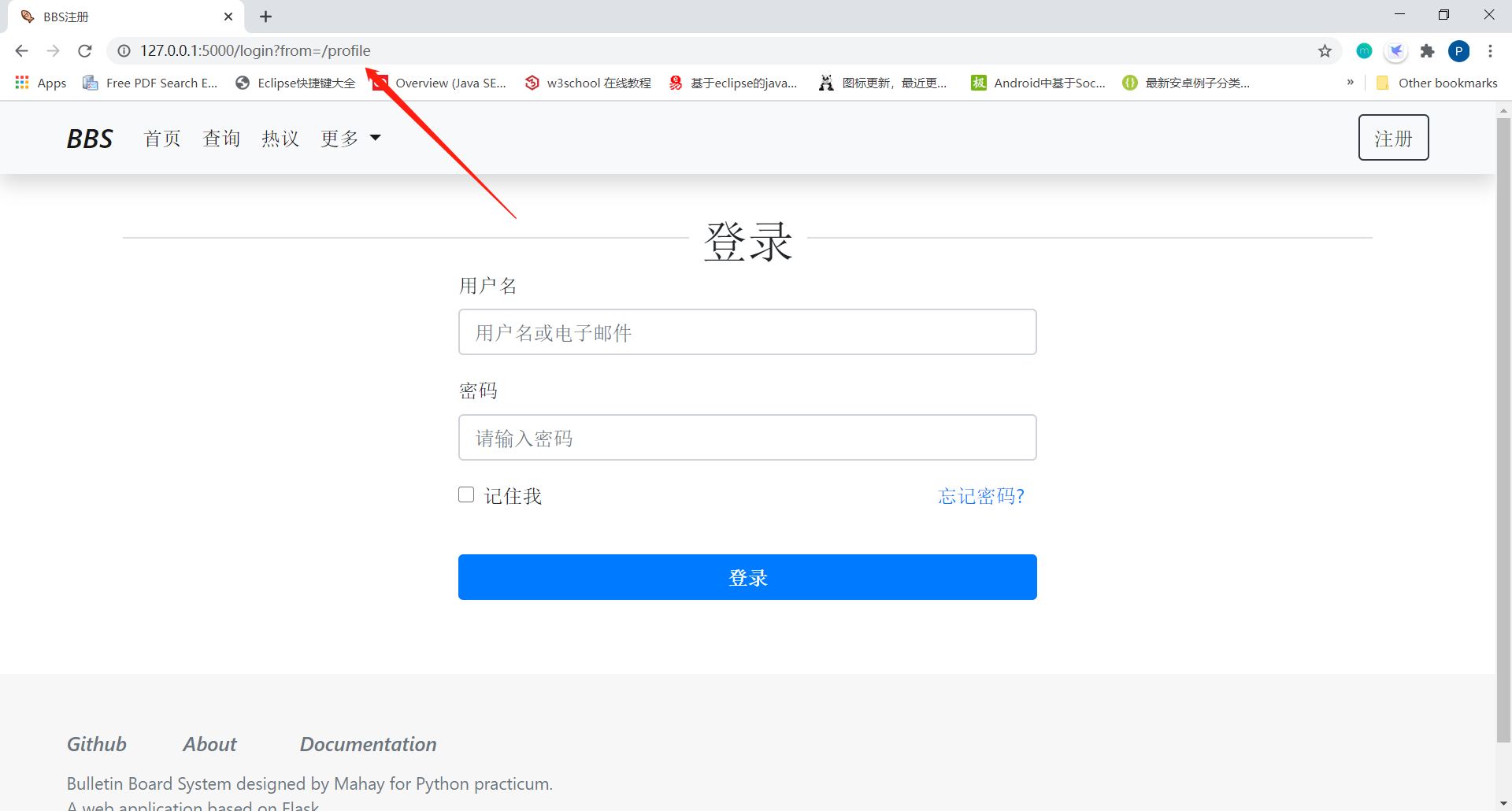


**图2.3.3 Controller按服务功能进行分类**

**三、功能介绍**

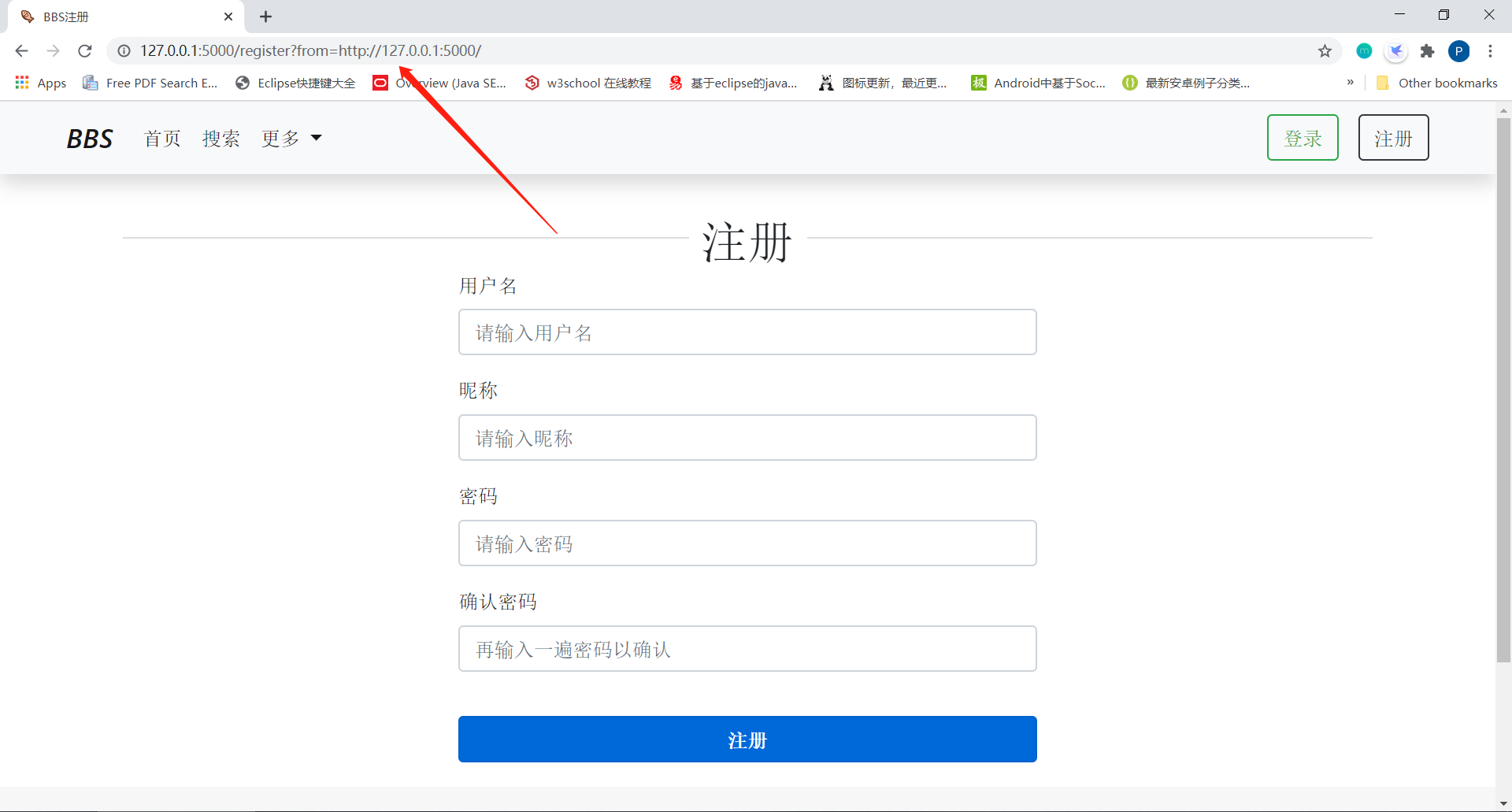
**3.1 登录与注册**

首先用户在未登录状态下访问任何需要登录的页面，将自动重定向到登录页面，在登录完成后将自动重定向回原来要访问的页面，原页面地址保存在查询参数from中，如图3.1.1所示：



**图3.1.1 重定向到登录页面功能**

在未登录状态，可以访问的页面右上角均有注册按钮，单击注册按钮将跳转到注册页面，注册完成后页面自动跳转到原来的页面，原页面地址保存在查询参数from中，如图3.1.2所示：



**图3.1.2 注册页面**

在注册时，需要输入用户名（用户名不可重复）、昵称、密码，并确认输入的密码，单击注册，会检查输入数据的合法性，若某项数据有误，将给出相应的提示，如图3.1.3和图3.1.4所示：



**图3.1.3 确认密码与密码不相同时给出的提示**



**图3.1.4 用户名重复给出的提示**

一旦提交了正确的数据，将提示注册成功，并且自动跳转到原来的页面。下一步，用户可以通过刚刚注册的账号进行登录，用户既可以在上文提到的自动重定向的登录页面登录，也可以点击页面右上角的登录按钮，在弹出的模态框中进行登录，如图3.1.5所示：



**图3.1.5 登录模态框**

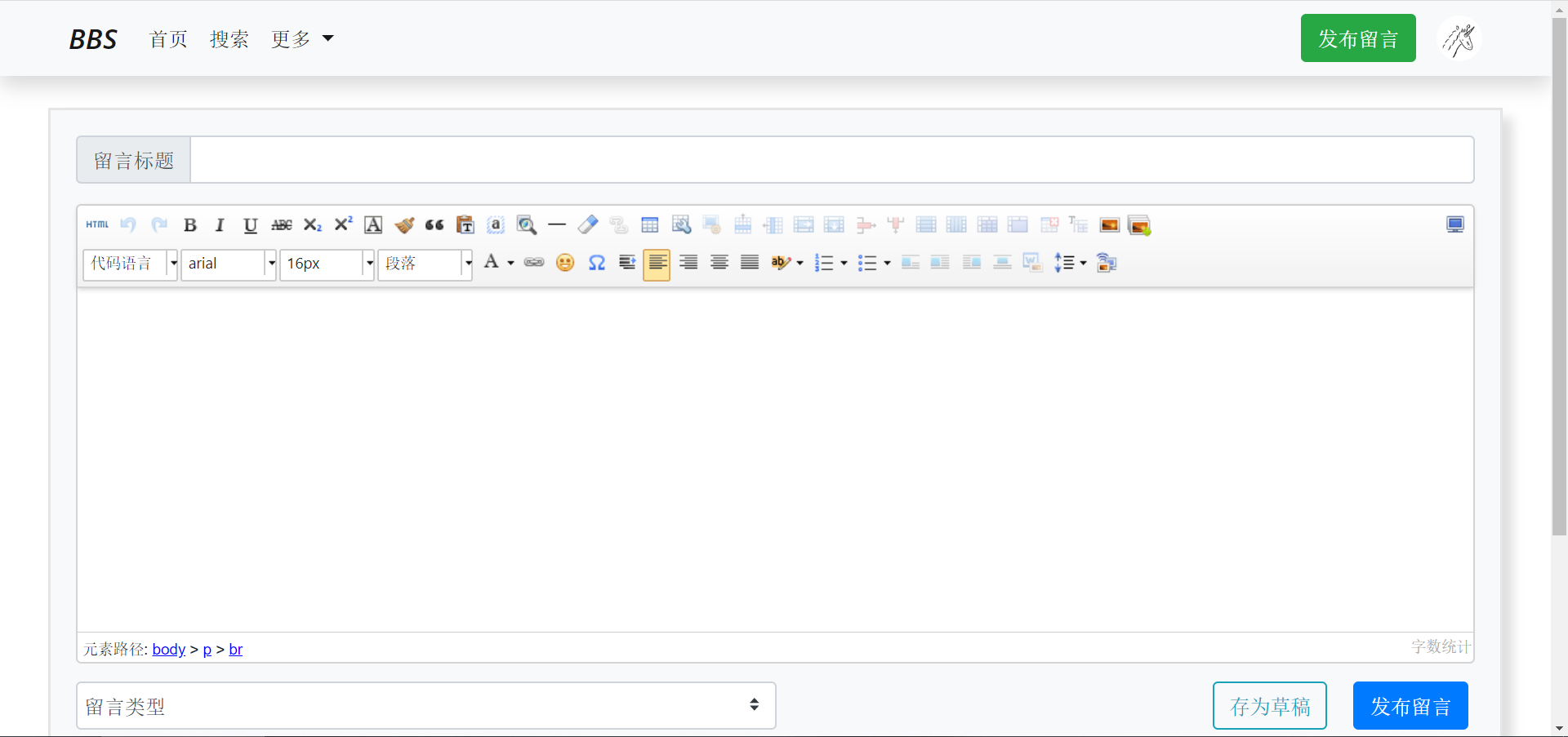
在登录模态框中，输入用户名和密码，单击登录尝试登录。如果输入数据非法或密码错误，同样会弹出提示，这里不太展示。另外，可以勾选“记住我”复选框，在下次访问页面时，将使用保存的Cookie自动进行登录验证，Cookie有效期30天，30天后需要再次手动登录刷新有效期限。登录成功后，页面给出提示，并自动刷新页面。登录后页面右上角将展示用户中心，如图3.1.6所示：



**图3.1.6 登录后右上角显示用户中心**

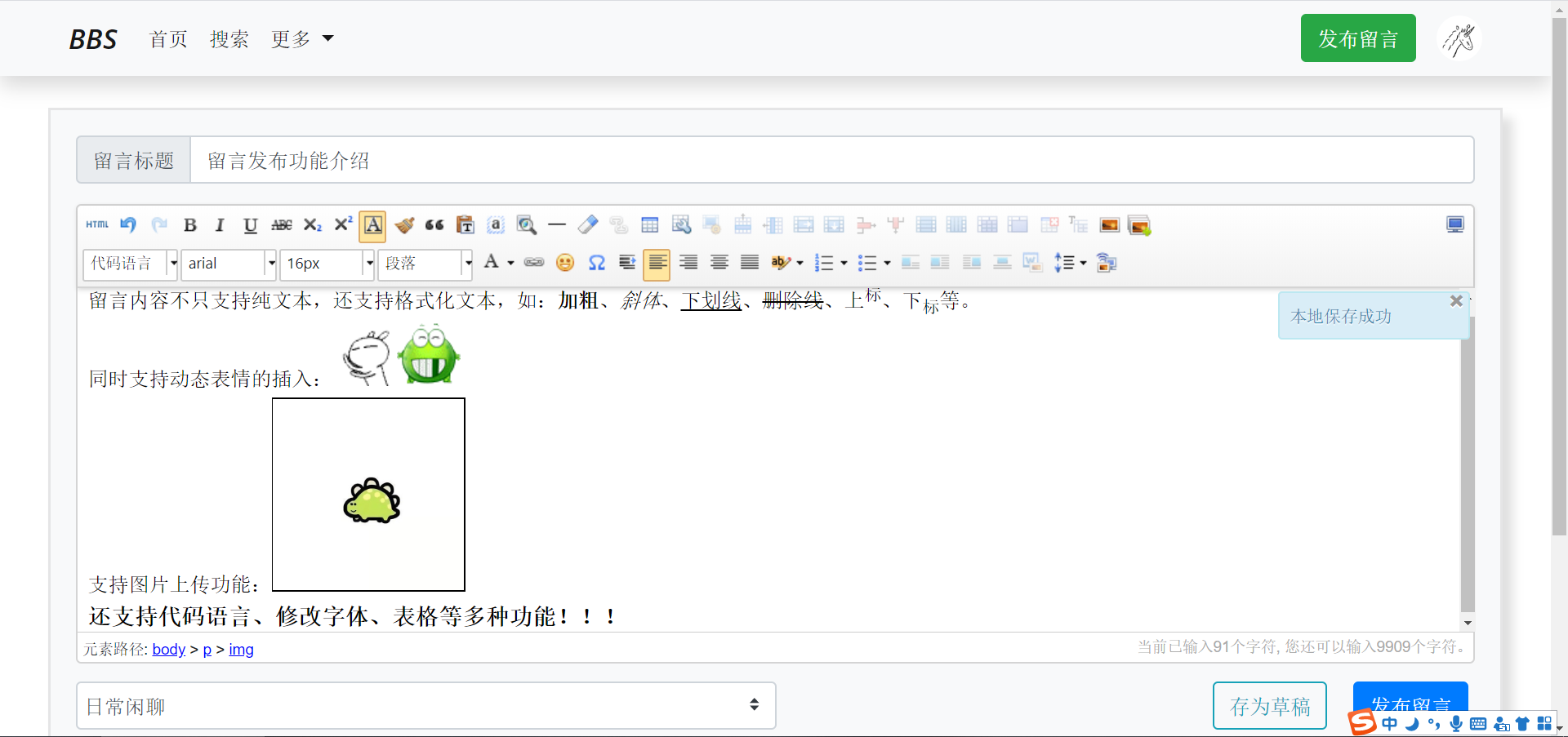
**3.2 留言发布**

用户在登录后，在页面右上角出现留言发布按钮，单击此按钮进入留言发布界面，如图3.2.1所示：



**图3.2.1 留言发布页面**

发布留言时，需要填写留言标题、留言内容，并选择留言类型。留言内容不只可以输入文字，而且支持多数富文本格式，如表情、动图、图片、表格以及各种格式化文本，如图3.2.2所示：



**图3.2.2 留言的富文本编辑功能**

用户编辑留言完成后，可选择存为草稿，待日后修改后再发布，也可以立即发布。单击发布留言按钮，若留言满足格式要求则发布成功，发布成功提示后自动跳转到留言阅读界面，以便作者观察效果是否符合预期，如图3.2.3所示：



**图3.2.3 留言发布成功后自动跳转到阅读页面**

**3.3 留言阅读与评论**

留言根据作者在发布时选择的类型分类显示，在首页，每个类型占据一个版块，版块显示目前最热门（根据回复数量排名）的前三条留言。版块顶部标明了留言分类名称和该分类下的留言数量，如图3.3.1所示：



**图3.3.1 首页显示热门留言**

用户可以直接点击留言标题进入留言阅读界面，也可点击版块上方或右侧边栏分类名称，进入该分类下了留言列表界面，如单击学习交流区，进入学习交流分类的留言列表，如图3.3.2所示：



**图3.3.2 分类留言列表界面**

留言列表每页显示10条，以发布时间降序排列，页面最下方显示了分页按钮，点击后页面显示下一页内容，如图3.3.3和图3.3.4所示：



**图3.3.3 页面底部显示分页按钮**



**图3.3.4 列表显示下一页内容**

点击留言标题进入留言阅读界面，界面与上文图3.2.3给出的相同。在阅读界面，显示留言作者、发布时间、阅读次数、回复次数等信息，同时可以在下方评论框对留言内容进行评论，评论发表后立即显示在留言下方评论区，如图3.3.5所示：



**图3.3.5 对留言进行评论**

对于自己发表的评论，用户可以进行隐藏/显示的控制，对于所有一级评论，用户可以对评论进行回复，单击回复按钮，将弹出一个回复框，发表后回复显示在对应评论下方，如图3.3.6和图3.3.7所示：



**图3.3.6 弹出的回复框**



**图3.3.7 回复显示在对应评论下方**

评论同样分页显示，项目使用AJAX异步加载渲染评论，评论换页时不需要页面整体刷新。

**3.4 个人信息管理**

用户将鼠标悬停在右上角头像上，可唤出个人信息管理菜单，如图3.4.1所示：



**图3.4.1 个人信息管理下拉列表**

单击注销按钮，可以退出登录当前账号，以便切换账号。单击个人中心按钮，进入个人信息管理页面，在该页面用户可以修改自己的昵称、头像和密码，如图3.4.2所示：

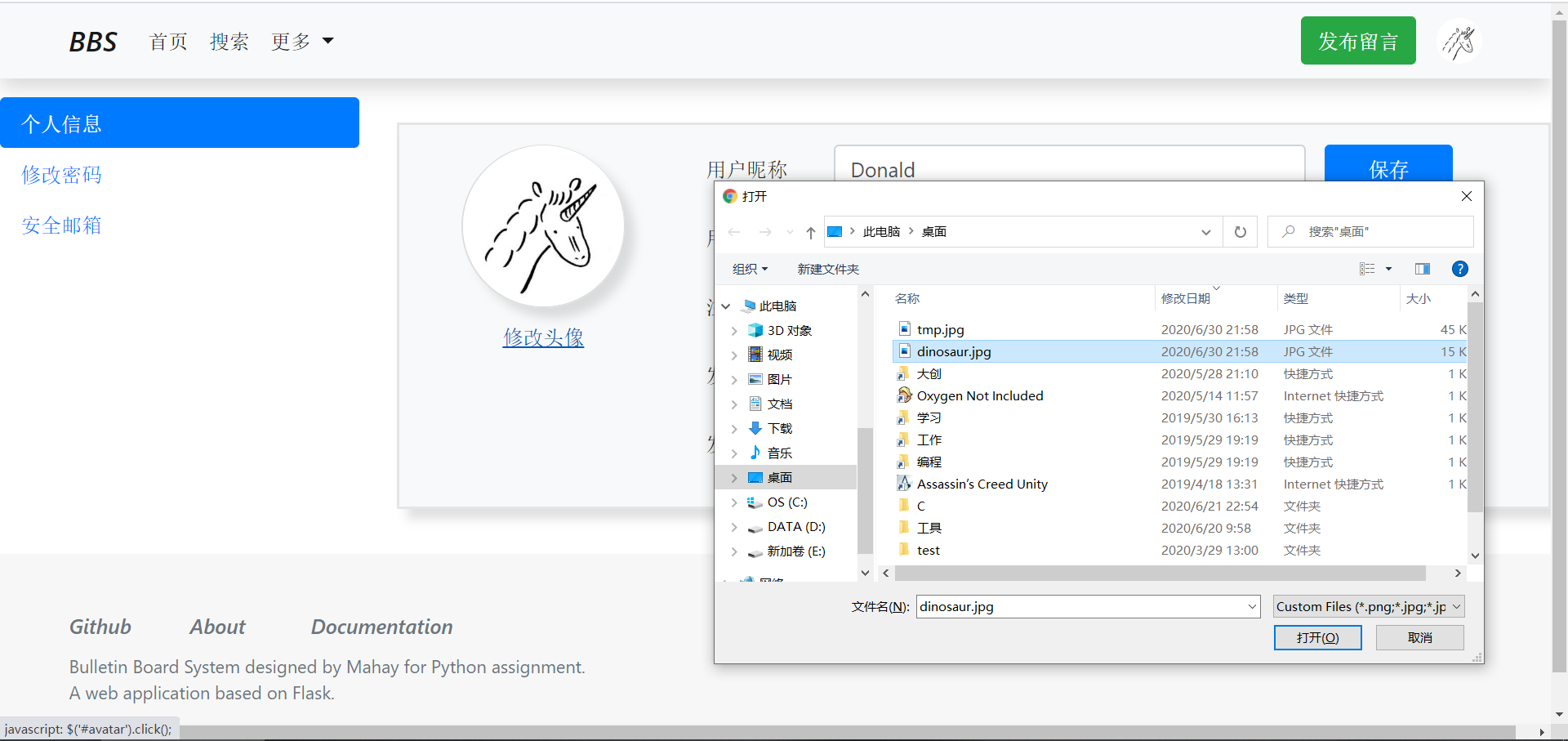


**图3.4.2 个人中心**

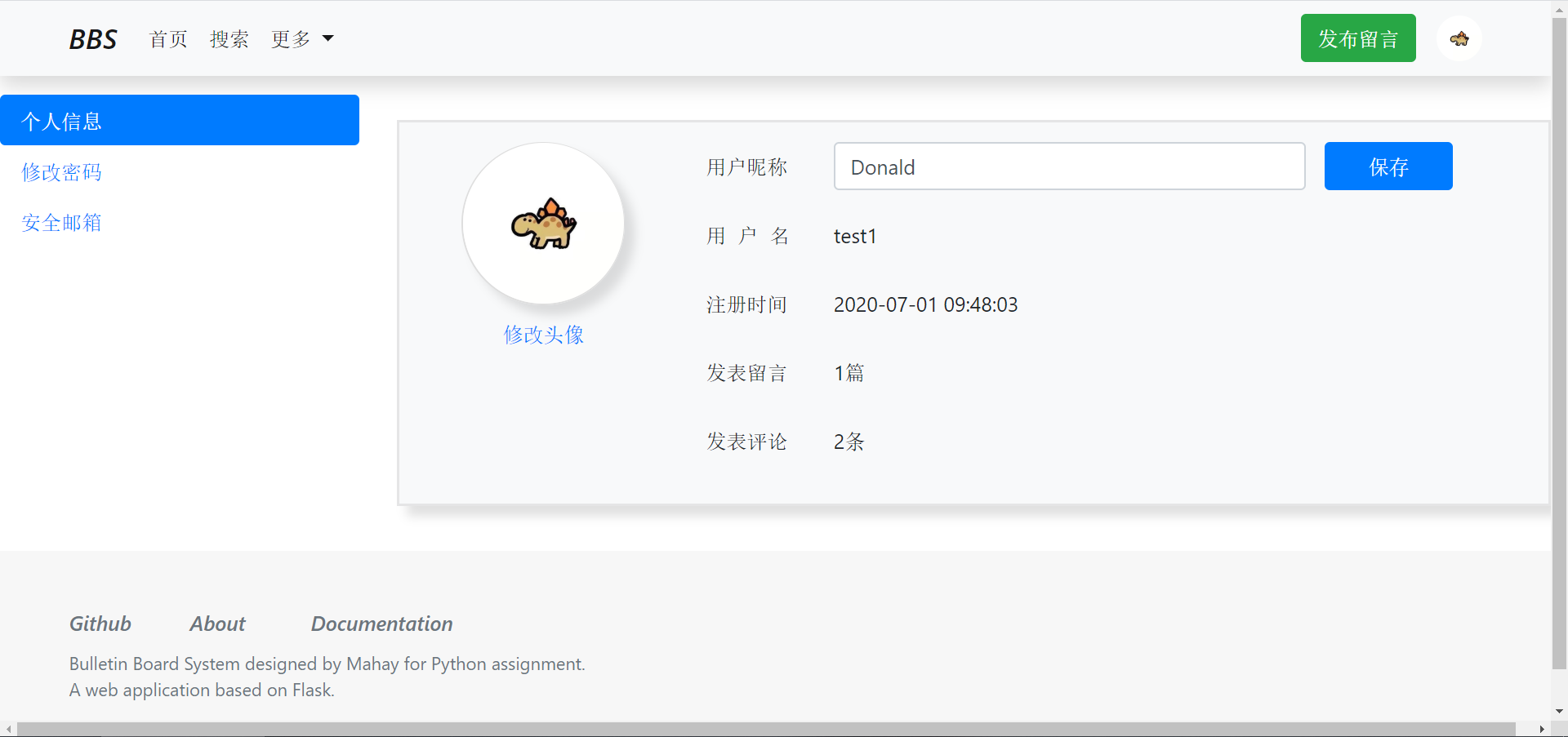
若需要修改昵称，在文本框输入新的昵称，点击保存按钮，若输入合法，则修改昵称成功，新昵称在用户昵称区显示。

若需要修改密码，进入修改密码Tab，输入原密码和新密码进行密码更改，下次进入网站时需要重新输入密码手动登录。

若需要修改头像，点击修改头像按钮，弹出文件选择对话框，选择要上传的图片，若图片格式正确，则头像上传成功，页面显示新上传的头像，如图3.4.3和图3.4.4所示：



**图3.4.3文件选择对话框**



**图3.4.4 更换的头像立即显示**

**3.5 个人留言管理**

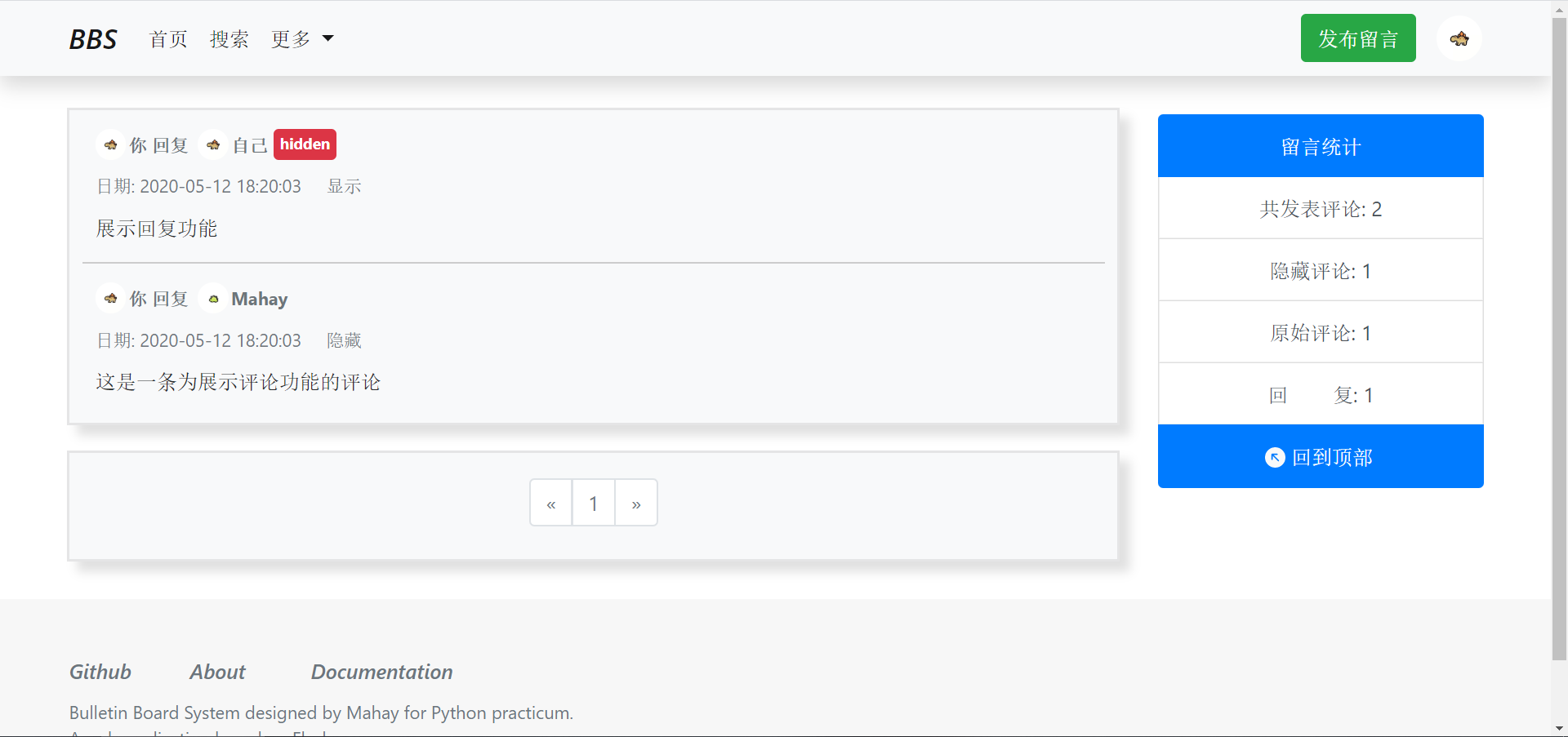
在上文图3.4.1所示的个人信息管理下拉列表中，还可以管理自己发布的留言（留言管理）、评论（评论管理），还可以看到别人给自己留言的评论和回复。

单击留言管理按钮，分页显示自己发布的所有留言，对于每条留言，可以进行隐藏/显示切换和修改操作。另外，如果一条留言是隐藏的，则会在标题前显示红色hidden标签，如果一条留言是草稿，则会在标题前显示黄色drafted标签，如图3.5.1所示：



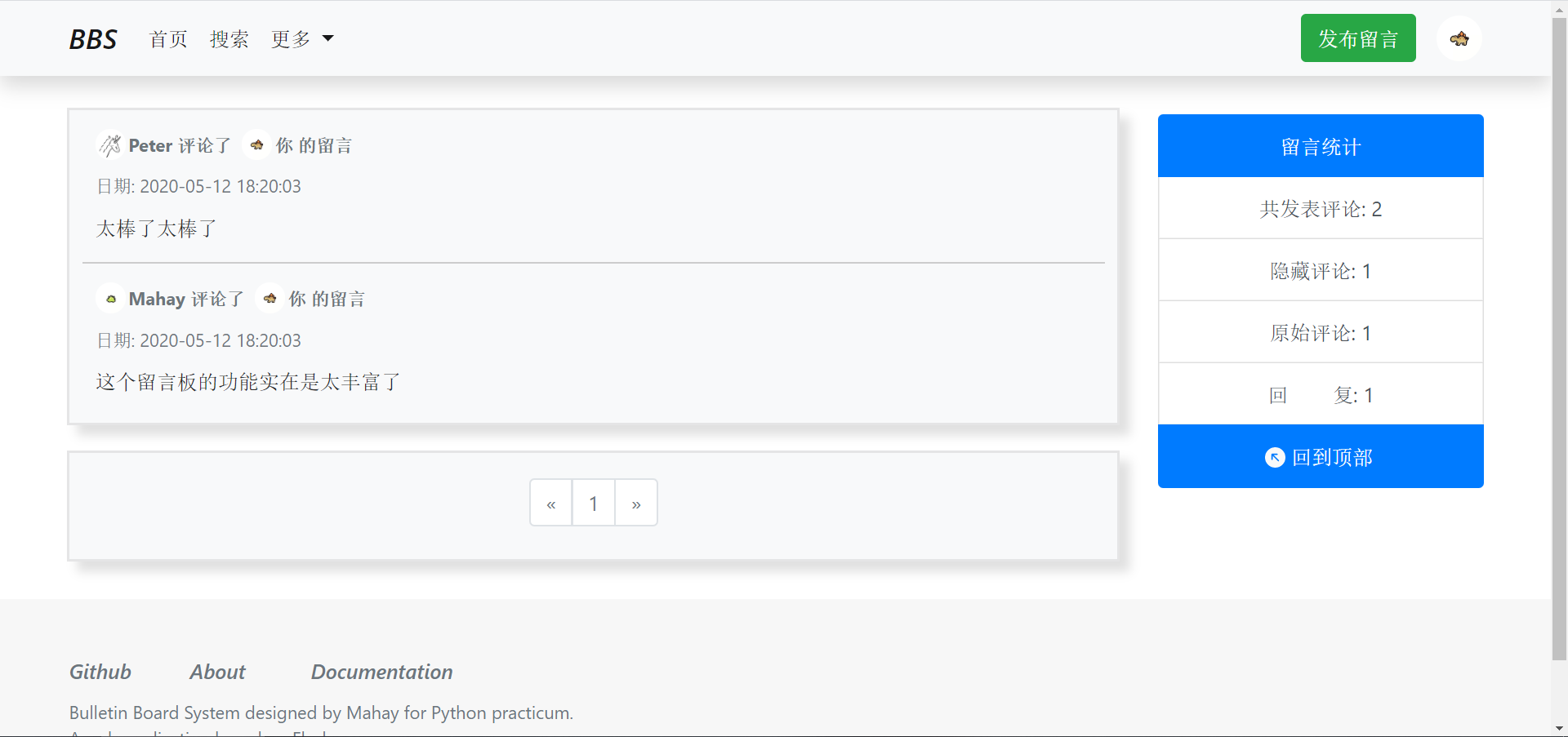
**图3.5.1 留言管理页面**

单击评论管理按钮，分页显示自己发表过的所有评论，并显示回复的用户头像和昵称，以及评论发表的日期等信息，同样，自己的评论可以进行隐藏/显示切换，另外点击评论内容可以跳转到相应留言阅读界面，如图3.5.2所示：



**图3.5.2 评论管理页面**

单击我的消息按钮，分页显示与我相关的评论和回复，点击评论内容可以进入相应留言阅读页面，如图3.5.3所示：



**图3.5.3 我的消息页面**

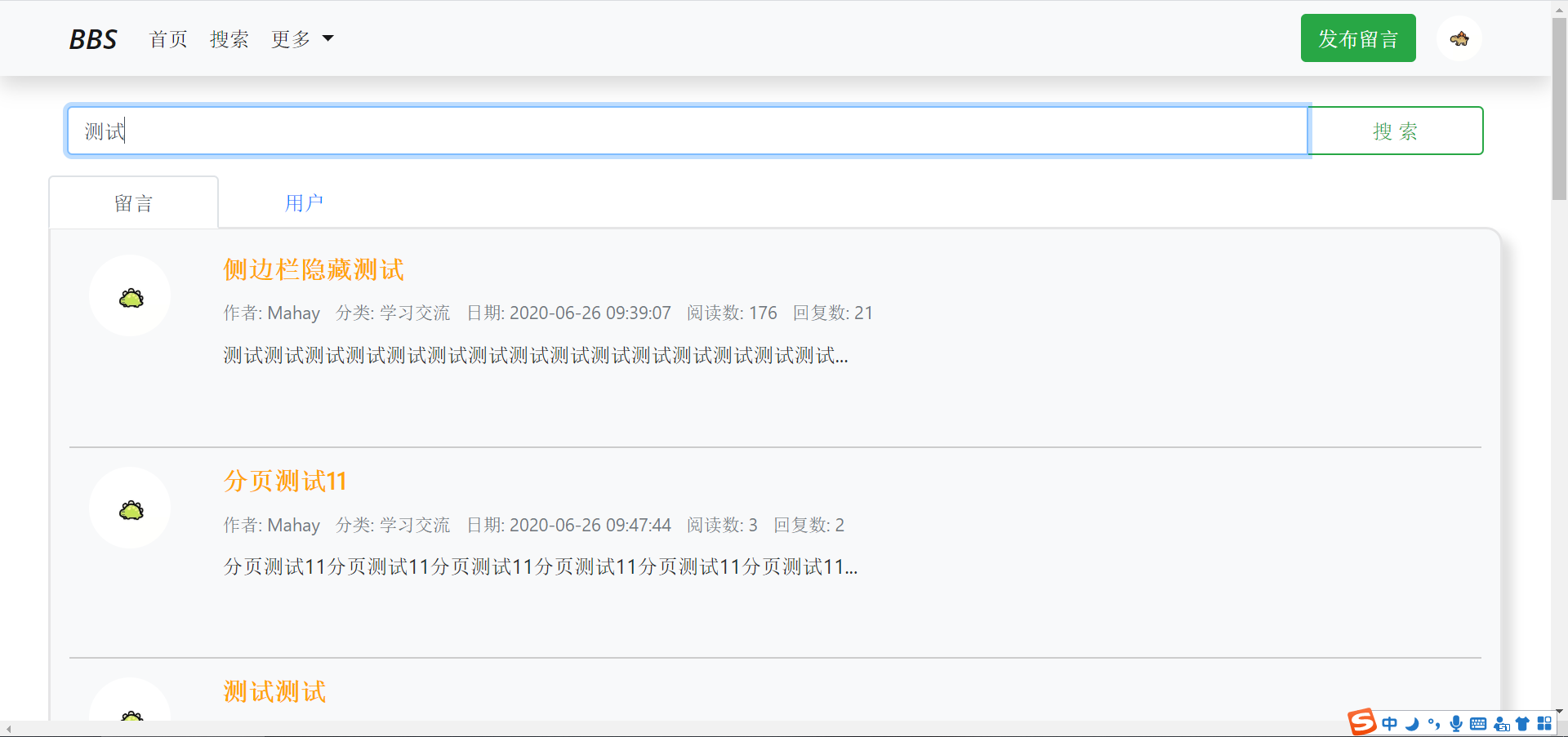
**3.6 全局搜索**

点击任何页面上方导航栏的搜索按钮，可以进入全局内容搜索页面，如图3.6.1所示：



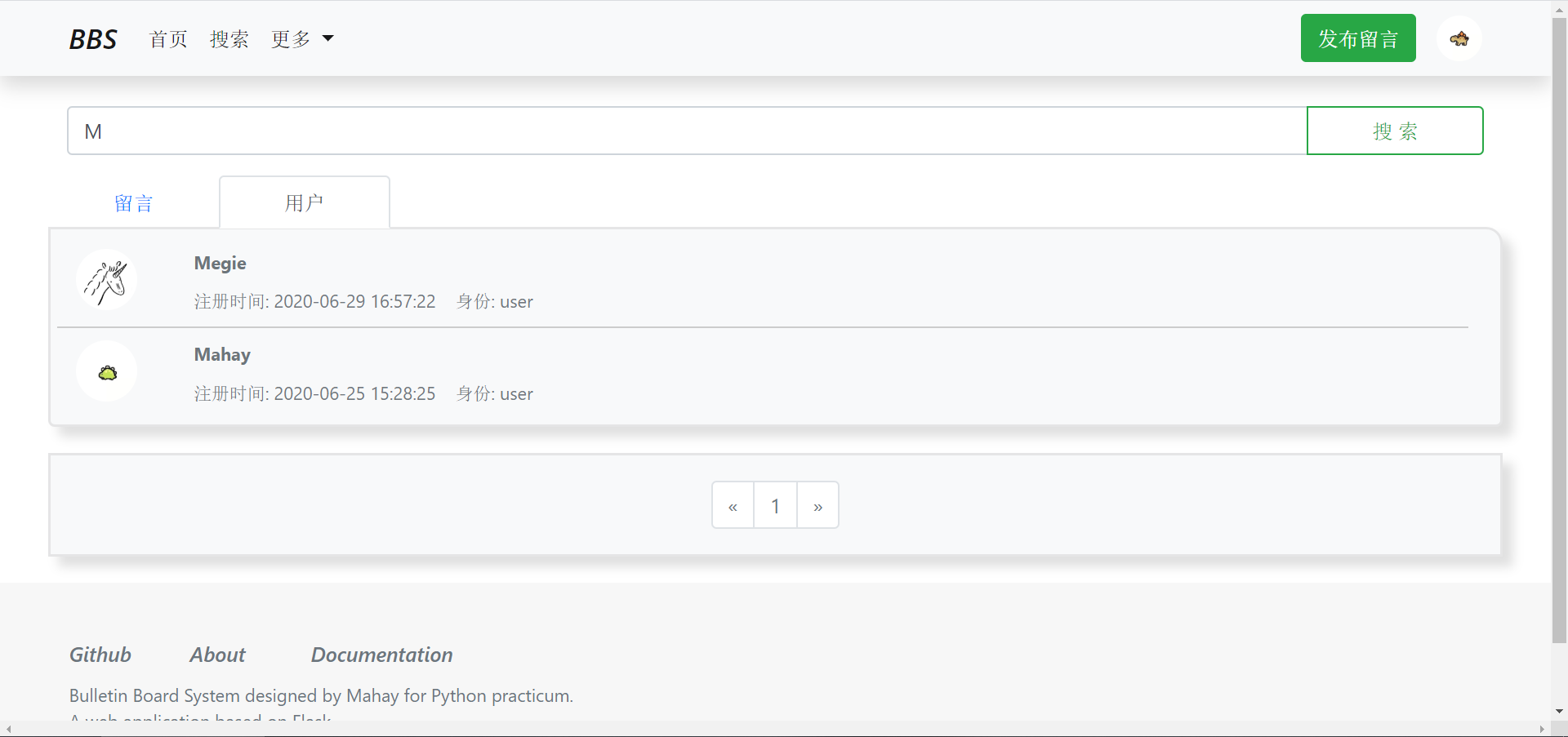
**图3.6.1 全局搜索页面入口**

搜索页面可以搜索留言，也可以搜素用户，默认搜索的是留言，都使用了模糊匹配的搜索方式。在搜索栏输入需要检索的关键词，敲击回车或点击搜索按钮进行留言搜索，将显示所有标题和内容包含关键字的留言，如图3.6.2所示：



**图3.6.2 搜索留言**

若需要搜索用户，则点击搜索栏下方用户Tab，将自动对关键字进行查询并显示结果，如图3.6.3所示：



**图3.6.3 搜索用户**

搜索结果同样使用AJAX技术异步分页加载，不需要刷新整个页面即可显示搜索结果。

**四、建议老师代码验证的步骤**

**4.1数据库建立**

项目下有建表脚本，**bbs-demo/sql/setup.sql**包含项目需要的所有表以及测试数据，在运行项目前请老师先使用source命令或相关工具建立表结构：

source 所在路径\setup.sql

**4.2 项目依赖**

项目后端代码依赖的库和框架保存在**bbs-demo/requirements.txt**中，请老师使用pip或类似命令安装项目依赖：

pip install -r requirements.txt

**五、总结**

**5.1 项目难点及解决方案**

**5.1.1** **难点**：系统较复杂，若功能分层不合理将各模块循环引用，降低代码可读性、可扩展性不说，还极有可能导致ImportError，当后期功能增多时，各功能模块耦合严重，ImportError发生的可能大大增加。

**解决方案：**查阅了几种常见的Web项目代码组织方式和几种软件架构模式，选择了Model-View-Controller的软件架构，再结合Flask提供的模块化工具Blueprint类和init\_app()等方法，成功解决了循环引用问题，并获得了较为清晰的代码层次结构。

**5.1.2 难点：**原生的HTML组件难以做到美观。

**解决方案：**搭配使用了CSS和Bootstrap框架，使用了许多Bootstrap提供的简约美观的组件，解决了页面美观问题。

**5.1.3 难点：**纯后端渲染的页面不够灵活，比如分页时必须刷新整个页面，且得到的页面不够动态，用户交互感不足。

**解决方案：**利用两天的时间学习了JS、CSS、JQuery和AJAX技术，分页时使用AJAX异步动态获取数据，再使用JQuery局部刷新页面，解决了分页问题；利用CSS配合JS以及JQuery提供的简单动画，实现了前端交互的动态效果，如上文图3.4.1所示鼠标悬浮显示下拉菜单的效果，提高了页面交互感。

**5.2 缺点和不足**

5.2.1 原计划增加绑定安全邮箱和找回密码的功能，但低估了项目的难度，时间不足，只是增加了相应的页面，但并没有后端功能支持

5.2.2 有些页面加载需要多次访问数据库，有些页面刷新一次就要访问一次数据库，造成性能的降低，可以考虑增加缓存和页面静态化功能

**5.3 收获**

首先，这次课程设计不仅是对Python基础知识的全面检验，也一定程度上促使了我接触新的技术，为了做出美观的界面，我学习了CSS和Bootstrap；为了给页面添加动态效果，我学习了JS和AJAX；为了简化JS编程，我又学习了JQuery。这让我加深了对前端代码的运行方式以及前后端的交互机理的理解，我想无论以后从事前端还是后端工作，这次课程设计的经历都会对我的工作有所帮助。

另外，独立完成一个实用的项目，是对我编程能力的一次巨大提升。通过这次机会，我把以前理解并不深刻的MVC架构模式用到了实际项目中，把Python的基础知识又翻出来重新巩固了一遍，锻炼到了对代码结构的组织能力和对编程知识的全面运用能力。而这是做小练习，编写一些支离破碎的代码片段，所锻炼不到的。

赵鹏飞

2020年7月1日

## 评分标准

1. 数据库设计规范性，合理性，数据对象的命名规范性 -----15分
2. 软件架构、代码文件结构组织的规范性，科学性 -----10分
3. 程序书写习惯，变量名的规范性，代码逻辑是否清晰等 -----10分
4. 系统功能的完成度，丰富性，最终系统的可用性 -----40分

可以自行扩展系统的必要功能，使系统的功能更强大注意，扩展的功能一定是相关的，适合实际应用场景的

1. 是否规定时间完成（根据代码提交的时间轴和版本） ----- 5分
2. 前端规范性及UI界面的美观性（爬虫程序项目可以无此项） -----10分
3. 软件系统开发设计说明文档-报告 -----10分