

目录

一、项目简介	1
二、程序的运行环境和安装步骤	1
2.1 运行环境	1
2.2 程序的组成部分	1
2.3 安装步骤	1
三、程序开发平台	2
3.1 代码行数	2
3.2 开发环境	2
四、程序功能说明	2
4.1 用户层面	2
4.1.1 注册/登录	2
4.1.2 修改个人信息及密码	6
4.2 校园模块	8
4.3 讨论模块	9
4.4 话题模块	11
4.5 消息模块	12
五、程序算法说明及面向对象实现技术方案	13
5.1 系统架构	13
5.2 中间层处理逻辑	14
5.3 redis 缓存的使用	14
六、技术亮点、关键点及解决方案	16
七、遇到的困难及解决方案	16
八、简要开发过程	17

“北理人”校园网页论坛项目

一、项目简介

一个简易的论坛系统“北理人”。该项目开发 IDE 为 IntelliJ IDEA，通过 Maven 构建，后台采用 Spring、SpringMVC、MyBatis，数据库采用 MySQL、Redis，前台用到 jQuery。

二、程序的运行环境和安装步骤

2.1 运行环境

JDK 1.8

2.2 程序的组成部分

业务代码和依赖包打包成一个 war，业务逻辑组成包括：用户管理模块，登陆注册管理(包含邮箱验证)，发帖与回复和点赞模块，校园模块，消息模块。依赖包主要有：springMVC，mybatis，redis，javax.mail。服务器：Tomcat，构建工具：maven

2.3 安装步骤

- 1) 安装 JRE/JDK 1.8。
- 2) 下载 tomcat 服务器. <http://tomcat.apache.org/download-80.cgi#8.5.27>
- 3) 安装 redis 并运行。(ubuntu: sudo apt-get update && sudo apt-get install redis-server)
- 4) 将程序文件复制到计算机上
- 5) 用 idea 打开程序所在目录
- 6) 刷新 maven 项目，等待 maven 下载依赖包

7) 配置 tomcat 服务器 (<https://www.cnblogs.com/Miracle-Maker/articles/6476687.html>)

8) 将 df.sql 文件中内容写入本地 mysql 数据库进行建表

9) 将 forum\src\main\java\com\limengting\util 中的 MyConstant.java 中的 IP 改成本机 IP, 即将下面一行中的 IP 地址进行修改;

```
public static final String DOMAIN_NAME = "http://192.168.1.104:8080/";
```

10) 将本地 mysql 的配置文件中的 sever 编码方式由默认的 latin1 改成 utf-8

11) 运行程序。

三、程序开发平台

3.1 代码行数

总代码约 9000 行, java 代码约 2000 行

3.2 开发环境

系统: ubuntu-17.10

IDE: IntelliJ idea version 2017.1.2 ultimate

四、程序功能说明

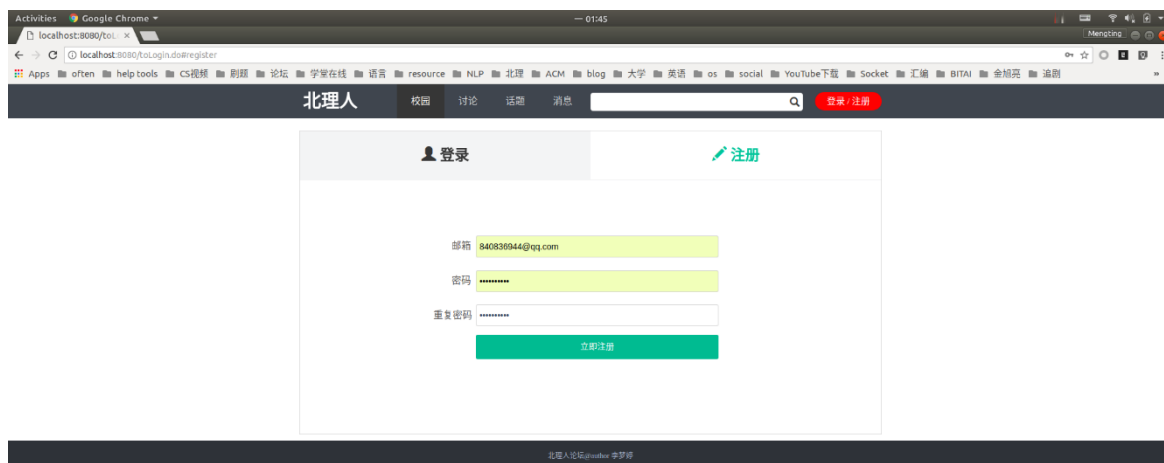
4.1 用户层面

4.1.1 注册/登录

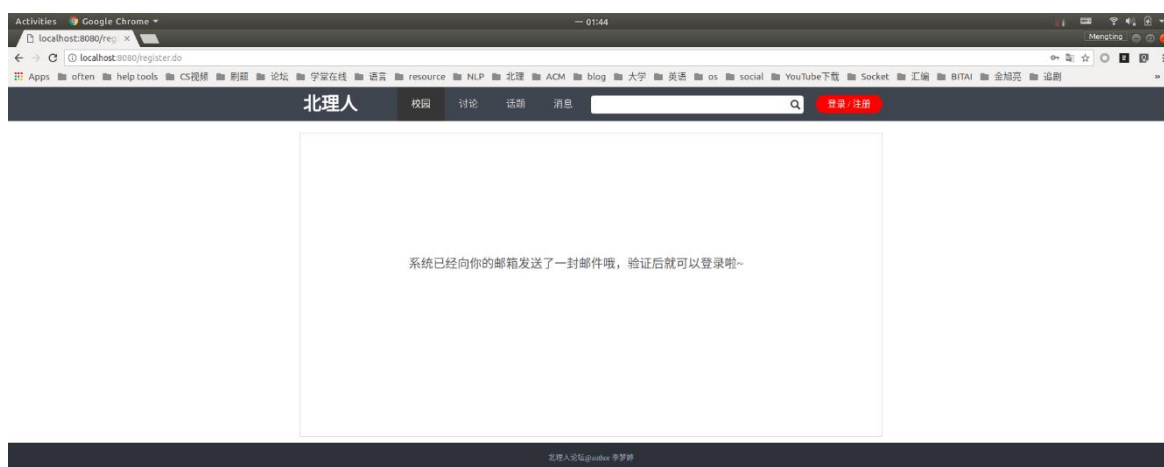
注册/登录功能实现较为完善, 注册为邮箱注册, 注册后由预设邮箱 (lmtsunniesina@sina.com) 发送给注册邮箱一封验证邮件, 点击验证链接则成功激活。注册考虑到了两次密码输入不同、邮箱已经被注册、邮箱格式不正确等情况, 登录考虑到了账号或密码错误、账号未被激活等情况。下面为具体功能截图:

“北理人”校园网页论坛项目开发文档

进入注册页面：

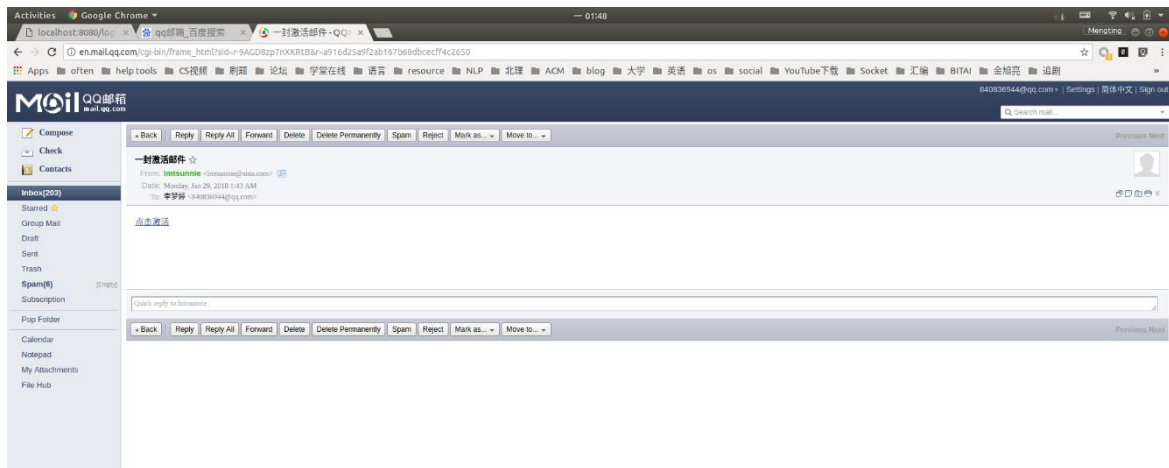


若邮箱格式正确且未被注册过，两次密码一致，则跳入已经发送邮件，等待激活页面：



打开邮箱则收到激活邮件如下：

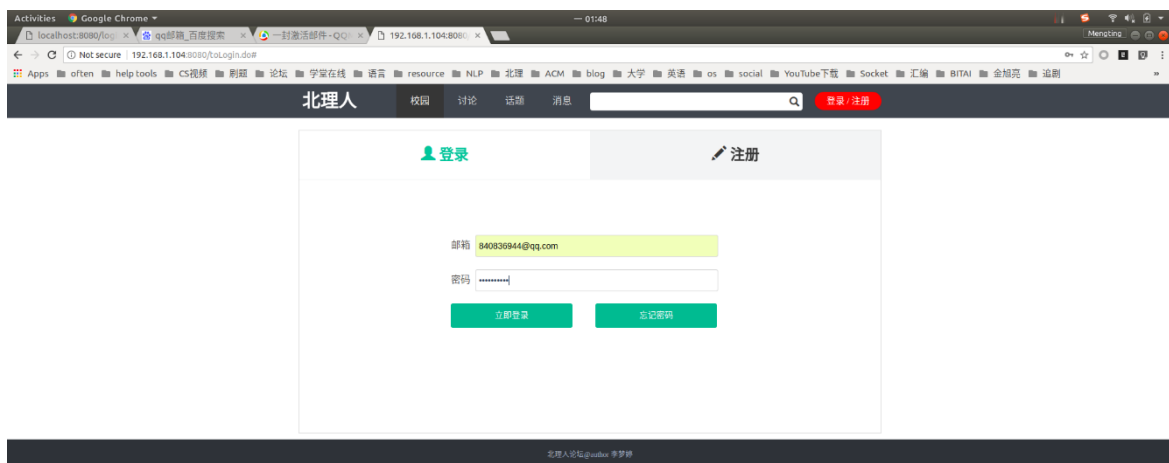
“北理人”校园网页论坛项目开发文档



点击链接则成功激活:



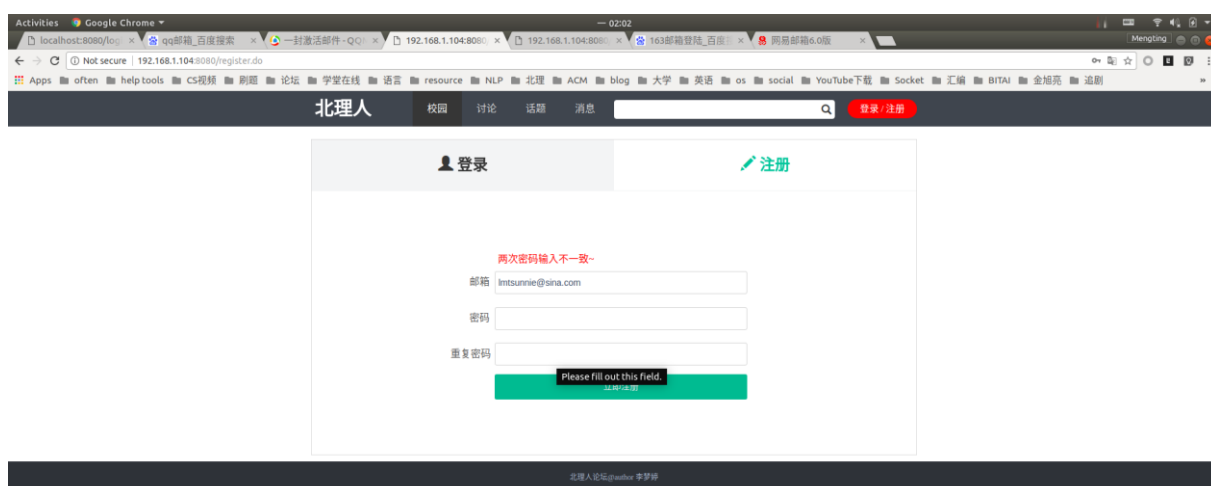
激活后登录:



“北理人”校园网页论坛项目开发文档

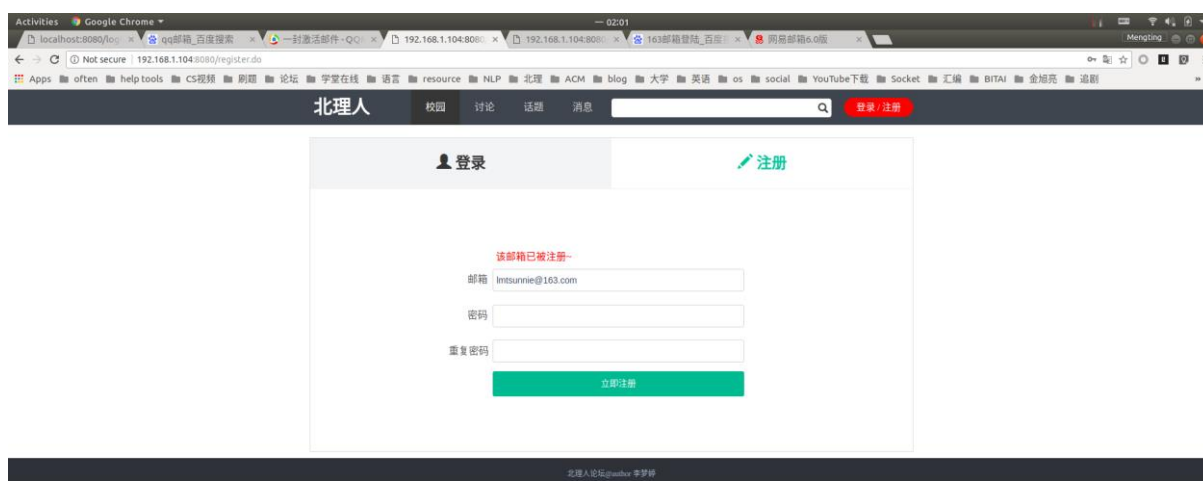
特殊情况处理：

(1) 两次输入密码不一致

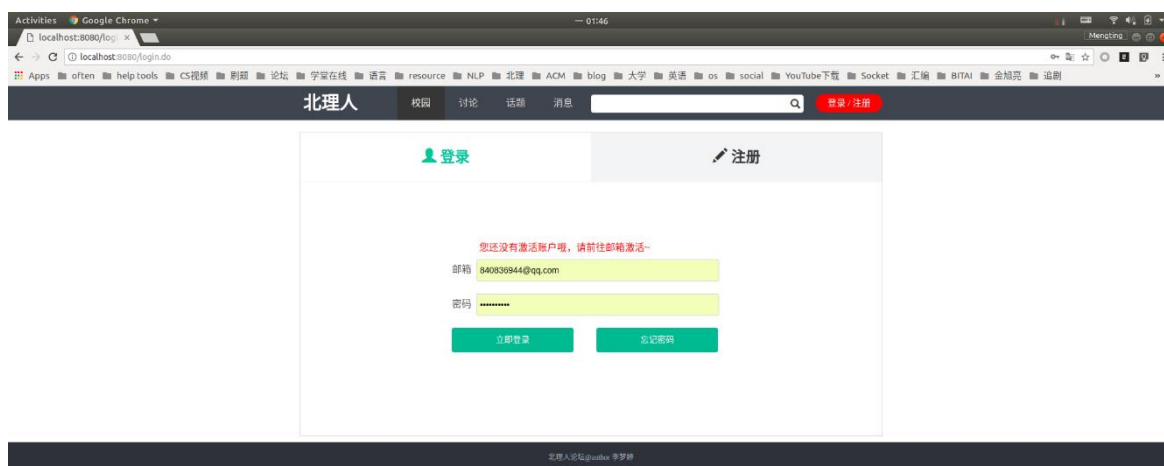


(2) 该邮箱已经被注册

“北理人”校园网页论坛项目开发文档



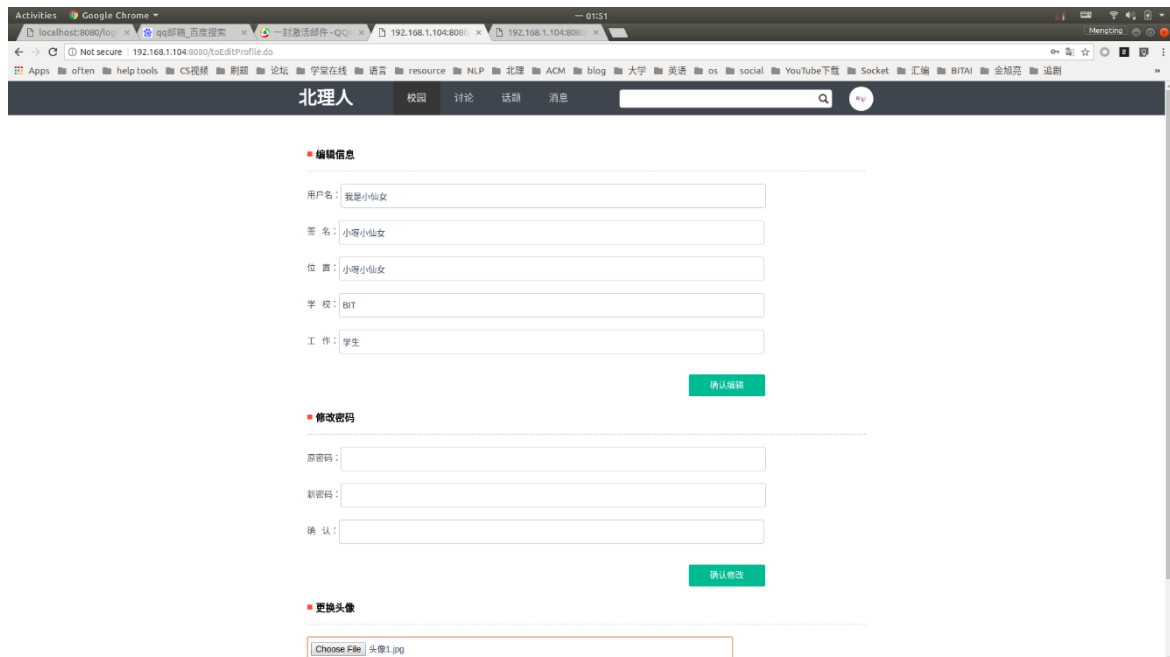
(3) 注册后没有激活就登录



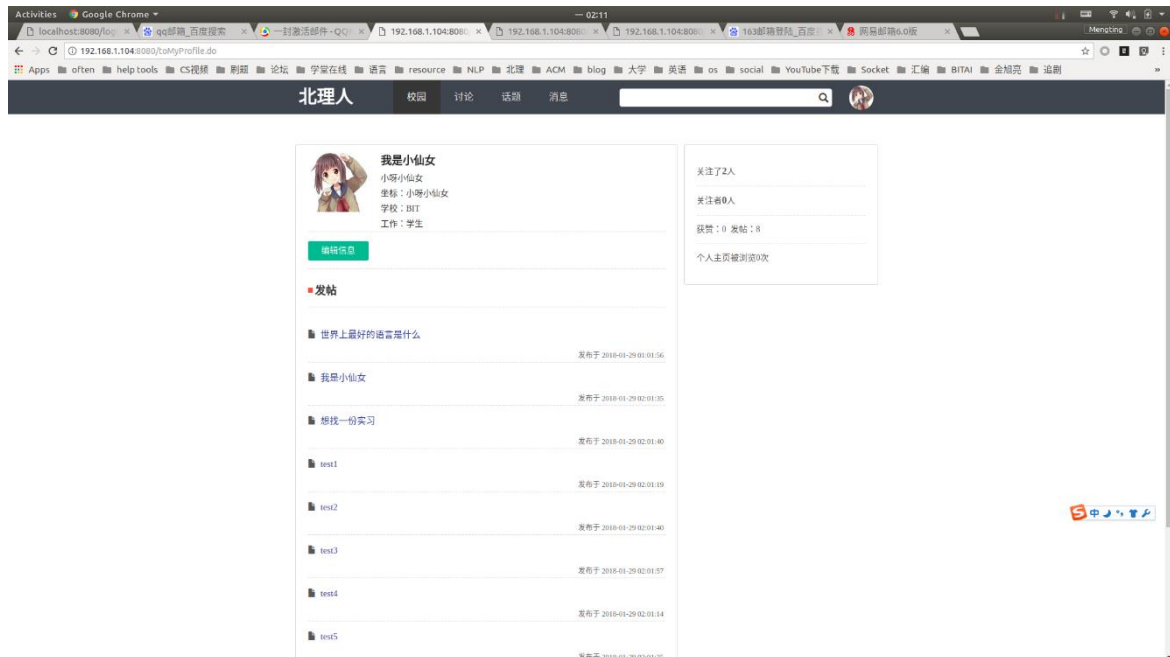
4.1.2 修改个人信息及密码

个人信息修改方面实现较为完善,支持修改用户名、签名等个人信息,修改密码,更换头像功能。

“北理人”校园网页论坛项目开发文档



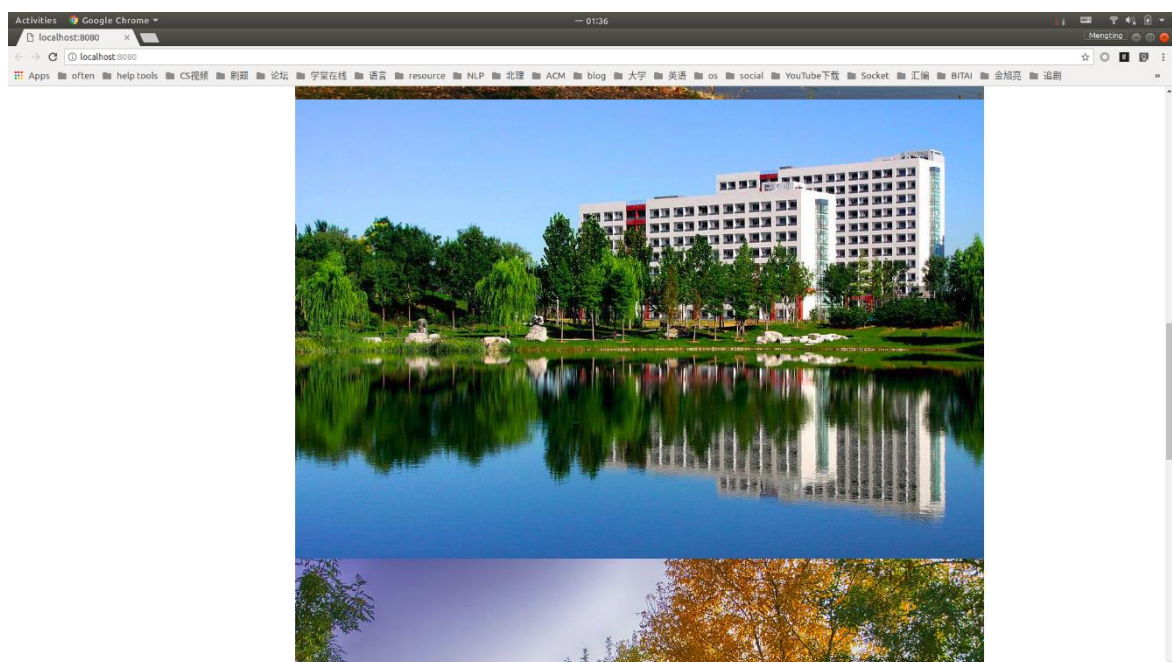
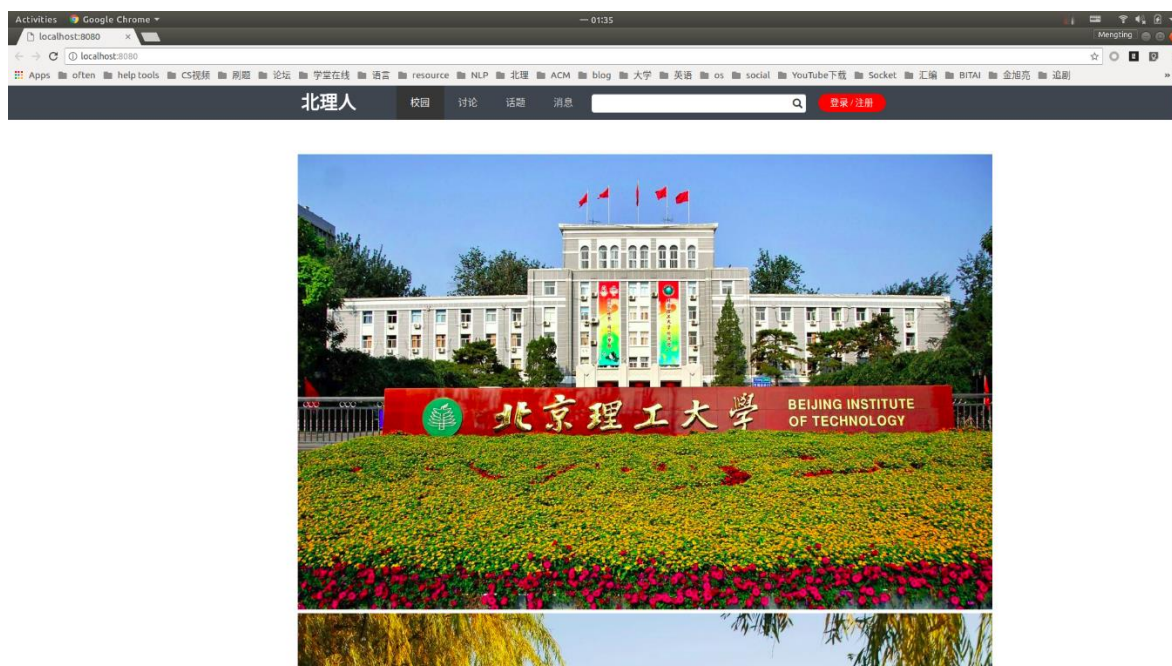
修改完成后：



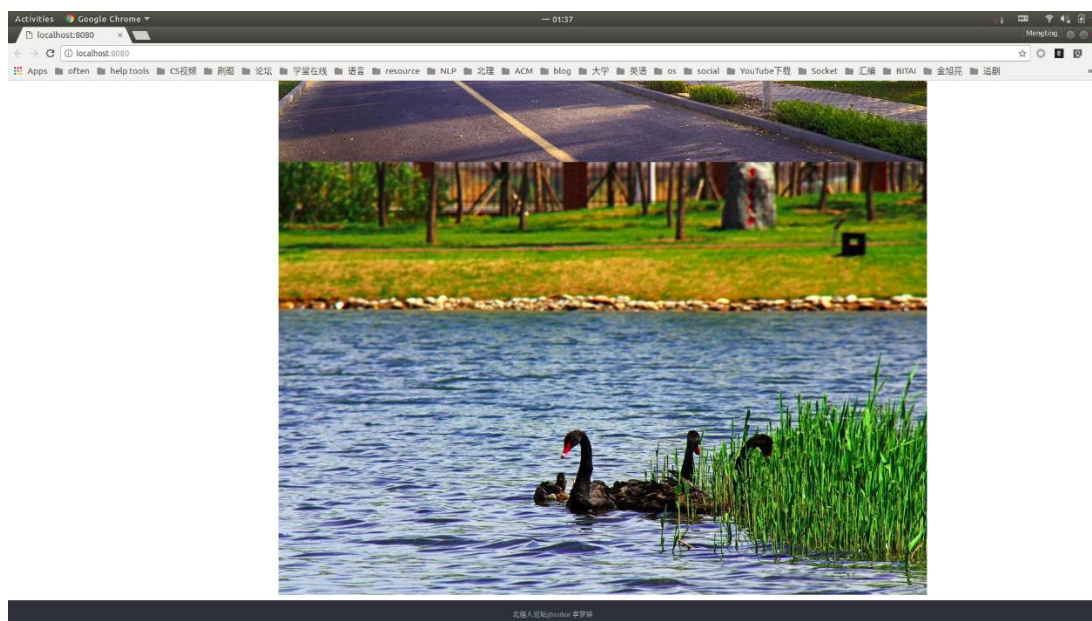
“北理人”校园网页论坛项目开发文档

4.2 校园模块

校园模块为“北理人”论坛的首页，如果在其他模块浏览，点击上方横条中的“北理人”或“校园”都能跳入校园景色主页进行浏览。



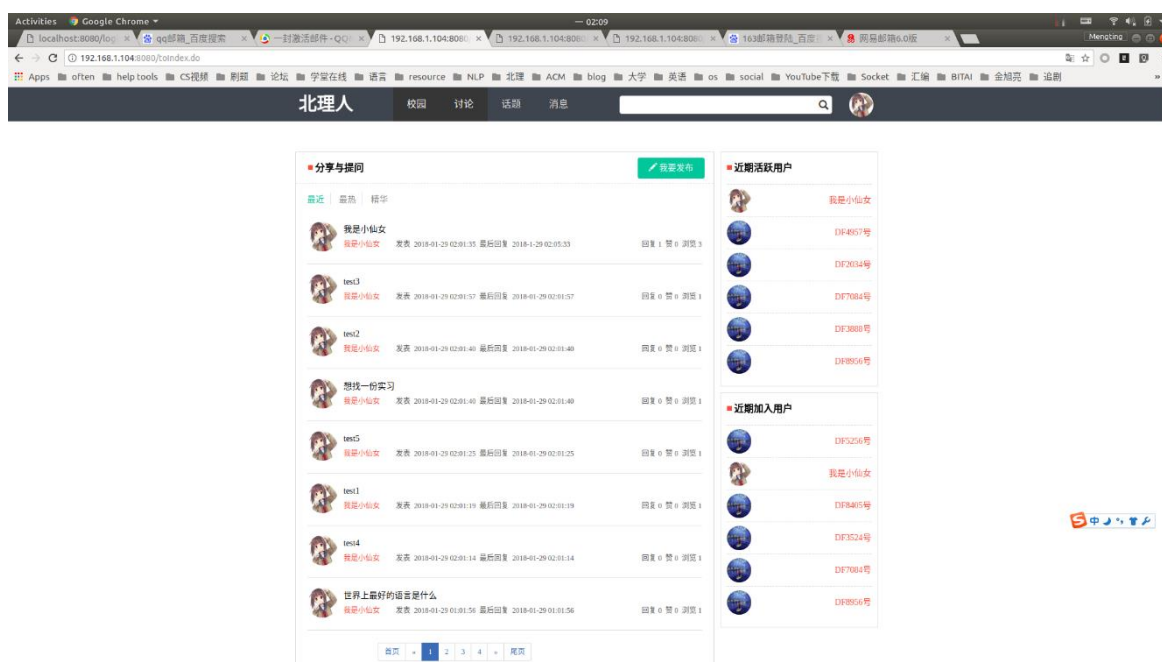
“北理人”校园网页论坛项目开发文档



4.3 讨论模块

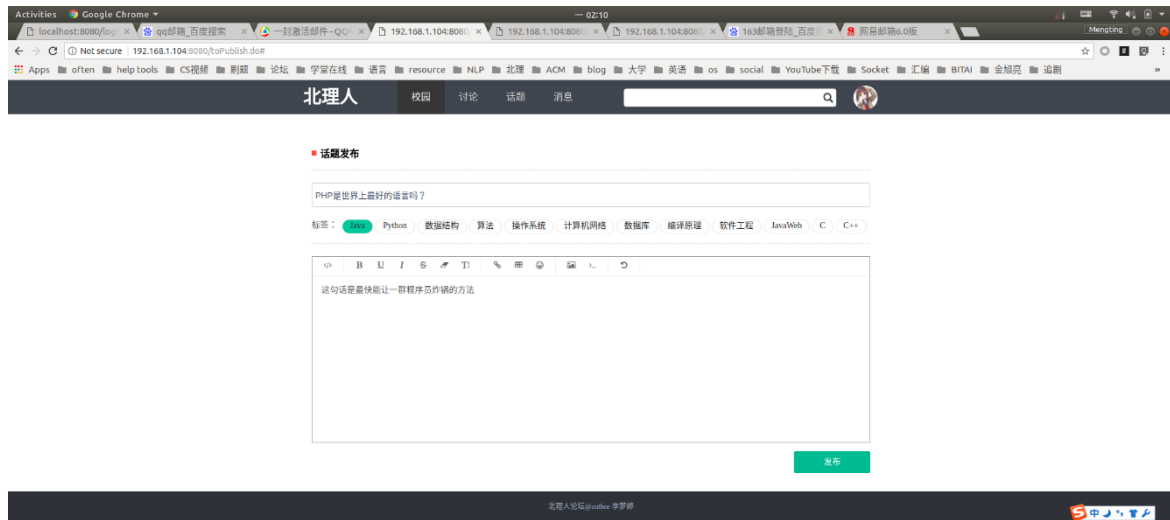
讨论模块实现了发帖、回复、点击发帖人或回帖人头像进入其主页、关注其他用户的功能。

讨论区主页：

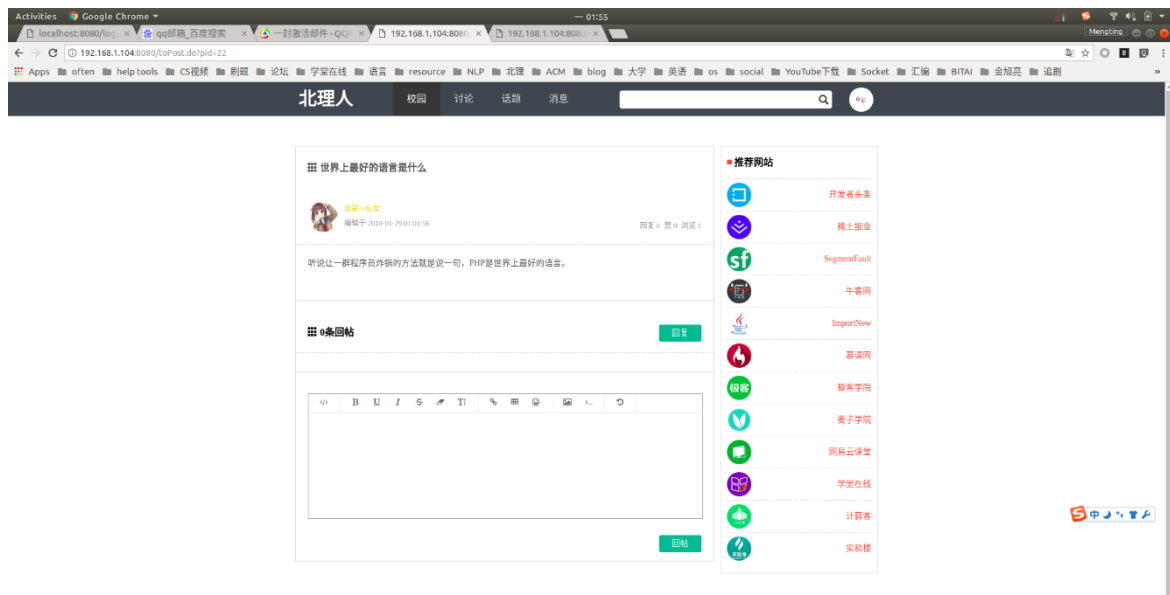


“北理人”校园网页论坛项目开发文档

讨论模块发帖：

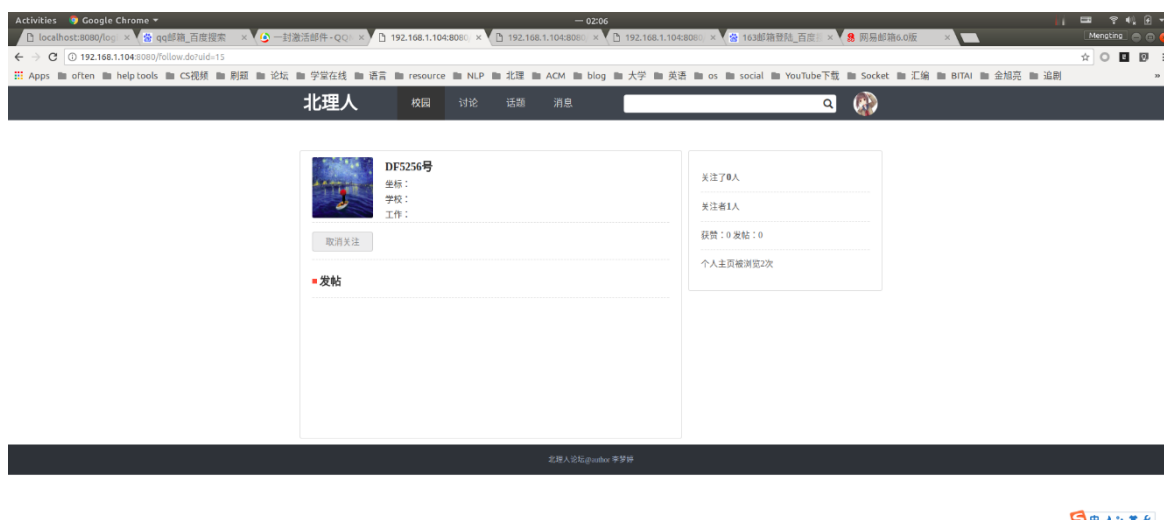


发布成功：



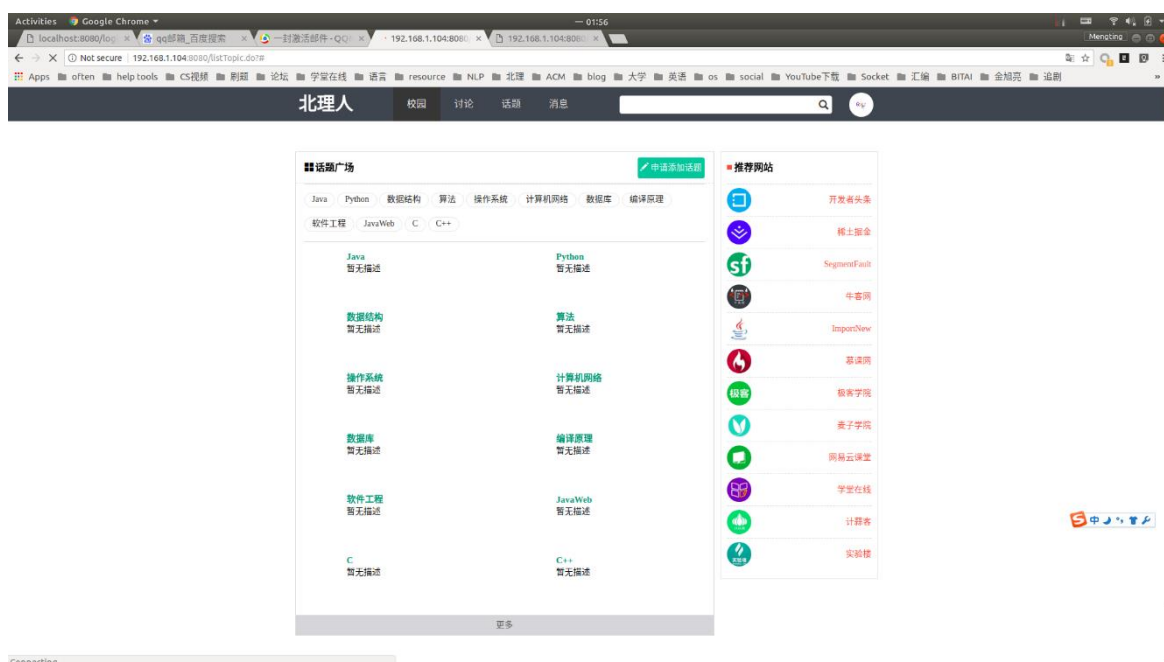
关注其他用户：

“北理人”校园网页论坛项目开发文档



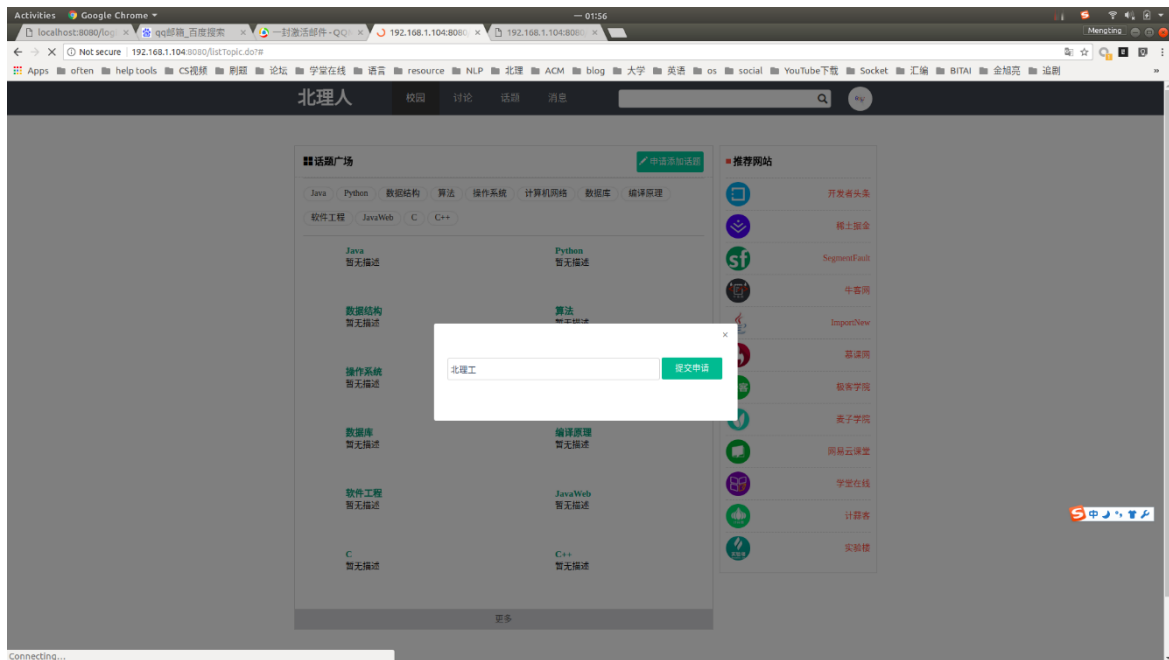
4.4 话题模块

由于时间原因，话题模块只搭建了界面框架，并未完全实现其功能。未实现的功能是当在讨论模块中发帖打上某个 tag 之后，在话题模块就能在相应 tag 的话题内找到该贴，依此将不同帖子按话题分类。已实现功能是添加新话题和右边的推荐网址。话题广场页面：



新增话题：

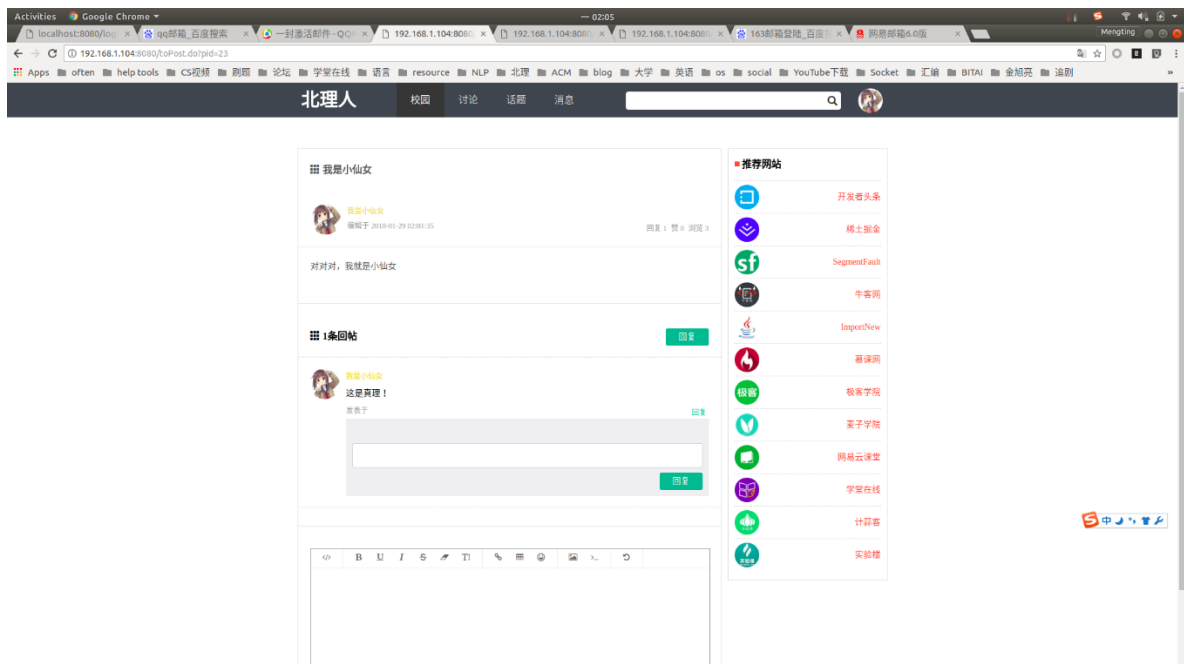
“北理人”校园网页论坛项目开发文档



4.5 消息模块

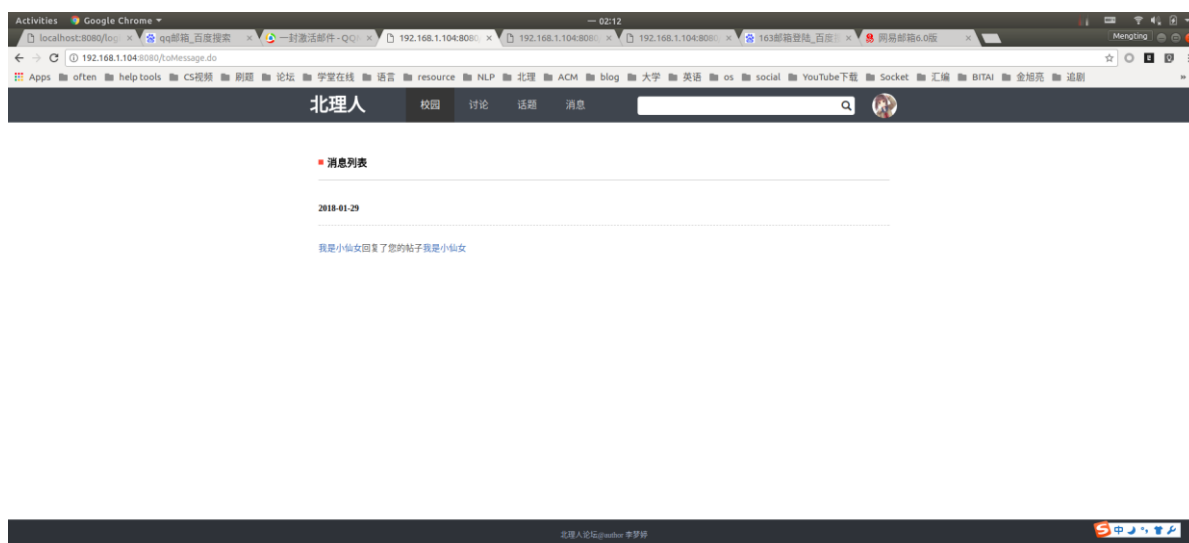
消息模块实现了当用户在讨论模块发帖被其他用户回复时，作为消息提醒。

以自己给自己回帖为例：



消息模块中的回帖提醒：

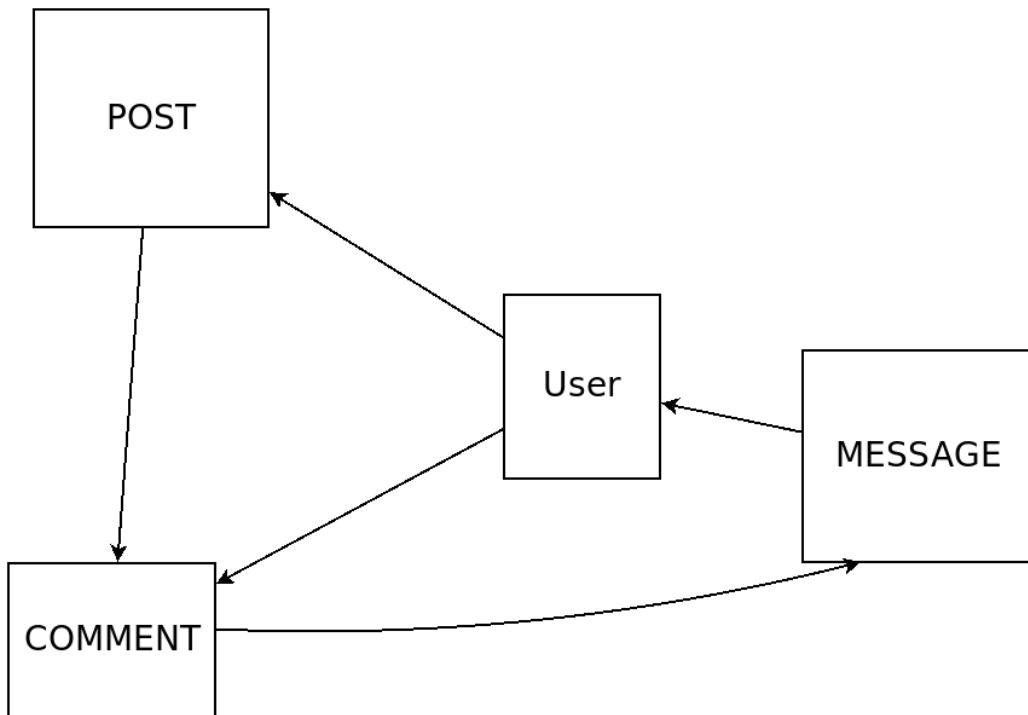
“北理人”校园网页论坛项目开发文档



五、程序算法说明及面向对象实现技术方案

5.1 系统架构

程序使用了标准的后端 MVC 开发模型，Controller 层主要有 UserController, PostController, TopicController, LoginController 等，主要是根据业务 url 不同，分发给不同的业务层 service 处理。service 层主要有 LoginService, MessageService, PostService, ReplyService, TopicService, UserService. Model 层主要有 User, Comment, Info, Message, Post, Reply, Topic, User 这些 javaBean 分别对应数据的一张表。同时，还有 javax.mail 和 redis 中间件来实现特定的业务逻辑。数据层交互用的 mybatis，每个 javaBean 对应一个 mapper 文件（可以用 mybatis 的 generate 插件直接根据数据库生成 model 和 mapper 文件）。



5.2 中间层处理逻辑

1. LoginInterceptor, 由于论坛有些功能游客无法访问, 需要用户登陆, 这个时候就需要有个登陆跳转机制, 这里我采用了 LoginInterceptor 模式, 当然也可以用 filter, 两者的比较可以参考:

<https://stackoverflow.com/questions/35686363/filters-vs-interceptors-in-web-application>

2. 邮件发送功能. 用户注册和忘记密码的时候, 需要有邮件验证, 增加流程上的完备性和和提高用户体验以及增加安全性。这里我们用到了 javax.mail 模块, 通过异步 task 的方式实现邮件的发送, 由于发送邮件依赖外部系统, 处理时间漫长而且不确定, 而注册需要及时返回, 为了提升用户体验, 这里我们用到了一个异步 task 来给用户发送通知邮件。

5.3 redis 缓存的使用

这里为什么要用 redis 呢, mysql 不是也能解决问题吗?

这个问题就涉及关系型和非关系型数据库的优缺点了。下面简单的介绍一下两

者的优缺点:

(1) 关系型数据库:

① 特点:

- 1、关系型数据库,是指采用了关系模型来组织数据的数据库;
- 2、关系型数据库的最大特点就是事务的一致性;
- 3、简单来说,关系模型指的就是二维表格模型,而一个关系型数据库就是由二维表及其之间的联系所组成的一个数据组织

② 优点:

- 1、容易理解:二维表结构是非常贴近逻辑世界一个概念,关系模型相对网状、层次等其他模型来说更容易理解;
- 2、使用方便:通用的 SQL 语言使得操作关系型数据库非常方便;
- 3、易于维护:丰富的完整性(实体完整性、参照完整性和用户定义的完整性)大大减低了数据冗余和数据不一致的概率;
- 4、支持 SQL,可用于复杂的查询。

③ 缺点:

- 1、为了维护一致性所付出的巨大代价就是其读写性能比较差;
- 2、固定的表结构;
- 3、高并发读写需求;
- 4、海量数据的高效率读写;

(2) 非关系型数据库:

① 特点:

- 1、使用键值对存储数据;
- 2、分布式;
- 3、一般不支持 ACID 特性;
- 4、非关系型数据库严格上不是一种数据库,应该是一种数据结构化存储方法的集合。

② 优点:

- 1、无需经过 sql 层的解析,读写性能很高;
- 2、基于键值对,数据没有耦合性,容易扩展;
- 3、存储数据的格式: nosql 的存储格式是 key, value 形式、文档形式、图片形

式等等，文档形式、图片形式等等，而关系型数据库则只支持基础类型。

③ 缺点

- 1、不提供 sql 支持，学习和使用成本较高；
- 2、无事务处理，附加功能和报表等支持也不好。

由于“北理人”论坛表结构关系复杂(虽然目前比较简单)，某些业务涉及多表查询，如果用户访问量多了，肯定会成为性能瓶颈，这里我们使用 redis，对于一些跨表聚合操作，我们可以将集合后的数值存储在 redis 缓存中，这样需要的时候我们就可以直接从缓存中获取而不需要去 Join 数据库表，大大提升了性能。比如某个帖子的点赞数量，某个用户的点赞总量(类似于知乎首页的点赞总量)，记录用户访问等。

六、技术亮点、关键点及解决方案

- 1、支持发送认证 email，用 javax.mail 接口实现，并且还是异步的解决方案。
- 2、支持上传头像。使用到了 commons-fileupload 实现文件的上传
- 3、用到了 redis 缓存。用到了最常用的 jedis 客户端支持。
- 4、支持基本的论坛功能。发帖回帖点赞等，类似于知乎。
- 5、第一次用 springMVC 和 mybatis，参照官方的文档，和网上搜索，总算解决了各种问题，瞬间成就感满满。
- 6、设计了一套数据库 UML，学以致用感觉很好。
- 7、用到了 MVC 架构。

七、遇到的困难及解决方案

(1) 用户注册后无法发送邮箱，spring 启动的时候报异常。

解决方案：多次网上搜索之后，最终发现是由于邮箱客户端代理需要在邮箱服务器打开 POP/SMTP 设置，然后才能发送邮箱。spring 启动异常是由于配置不正确，最终通过启动日志定位到了问题，并成功解决。

(2) 用户登录后修改用户名、签名等字符时出现乱码，修改后显示均为“?”号。

解决方案：mysql 字符编码问题，在 mysql 服务器的配置文件(/etc/mysql/mysql.conf.d)中加入以下两行把 mysql server 的编码方式由默认的 latin1 改成 utf8：

```
character_set_server = utf8
```

```
init_connect = 'SET NAMES utf8'
```

八、简要开发过程

12 月初	确定选题
12 月中旬	查阅资料
12 月~1 月 27 号	code, debug, code, debug, code, debug
1 月 27 号	对程序进行集成测试
1 月 28 号	程序开发工作完毕，编写及整理文档