**东北大学计算机科学与工程学院软件工程课程**



**“LSIC社区”**

**大学生学习生活信息交流网站**

**系统设计文档**

**小组git访问地址：**

[**https://github.com/Software-engineering-team-LSIC**](https://github.com/Software-engineering-team-LSIC)

**“LSIC社区” 实现及测试小组**

|  |  |
| --- | --- |
| **总组长：** | **黄 菊** |
| **实现组长：** | **黄 菊** |
| **测试组长：** | **徐建伟** |
| **组员：** | **刘 畅** |
|  | **侯宝玉** |
|  | **赵 肖** |

**指导教师**：**高岩**

2020年5月

**目 录**

[1 系统实现 1](#_Toc39859079)

[1.1人员安排 1](#_Toc39859080)

[1.2 系统通用类实现 1](#_Toc39859081)

[1.2.1 数据库通用类实现 1](#_Toc39859082)

[1.2.2 其它通用类实现 12](#_Toc39859083)

[1.3 主页面模块实现 12](#_Toc39859084)

[1.3.1 主页面模块实现简介 12](#_Toc39859085)

[1.3.2 主页面模块相关类实现 14](#_Toc39859086)

[1.3.3 主页面模块文件及跳转关系 17](#_Toc39859087)

[1.3.4主页面模块程序流程 18](#_Toc39859088)

[1.3.5 主页面模块实现界面 19](#_Toc39859089)

[1.4 分类专区模块实现 21](#_Toc39859090)

[1.4.1 分类专区模块实现简介 21](#_Toc39859091)

[1.4.2 分类专区模块相关类实现 22](#_Toc39859092)

[1.4.3 分类专区模块文件及跳转关系 24](#_Toc39859093)

[1.4.4 分类专区模块程序流程 25](#_Toc39859094)

[1.4.5 分类专区模块实现界面 26](#_Toc39859095)

[1.5 查看与回复帖子模块实现 31](#_Toc39859096)

[1.5.1 查看与回复帖子模块实现简介 31](#_Toc39859097)

[1.5.2 查看与回复帖子模块相关类实现 32](#_Toc39859098)

[1.5.3 查看与回复帖子模块文件及跳转关系 36](#_Toc39859099)

[1.5.4查看与回复帖子模块程序流程 37](#_Toc39859100)

[1.3.5 查看与回复帖子模块实现界面 38](#_Toc39859101)

[1.6 用户登录和退出模块实现 39](#_Toc39859102)

[1.6.1 用户登录和退出模块实现简介 39](#_Toc39859103)

[1.6.2 用户登录和退出模块相关类实现 40](#_Toc39859104)

[1.6.3 用户登录和退出模块文件及跳转关系 42](#_Toc39859105)

[1.6.4 用户登录和退出模块程序流程 43](#_Toc39859106)

[1.6.5 用户登录和退出模块实现界面 44](#_Toc39859107)

[1.7 实现过程中的需求和设计变动情况说明 45](#_Toc39859108)

[1.8 实现小结 46](#_Toc39859109)

[2 系统测试 48](#_Toc39859110)

[2.1人员安排 48](#_Toc39859111)

[2.2 系统通用类测试 48](#_Toc39859112)

[2.2.1 数据库通用类测试 48](#_Toc39859113)

[2.2.2 网络连接通用类测试 49](#_Toc39859114)

[2.2.3 系统通用类测试报告 50](#_Toc39859115)

[2.3生活模块测试 51](#_Toc39859116)

[2.3.1 生活模块白盒测试用例 51](#_Toc39859117)

[2.3.2 生活模块黑盒测试用例 52](#_Toc39859118)

[2.3.3 生活模块测试报告 52](#_Toc39859119)

[2.3.4 生活模块调试过程 53](#_Toc39859120)

[2.4 学习模块测试 53](#_Toc39859121)

[2.4.1学习模块白盒测试用例 53](#_Toc39859122)

[2.4.2 学习模块黑盒测试用例 54](#_Toc39859123)

[2.4.3 学习模块测试报告 54](#_Toc39859124)

[2.4.4 学习模块调试过程 55](#_Toc39859125)

[2.5 资源模块测试 55](#_Toc39859126)

[2.5.1 资源模块白盒测试用例 56](#_Toc39859127)

[2.5.2 资源模块黑盒测试用例 57](#_Toc39859128)

[2.5.3 资源模块测试报告 57](#_Toc39859129)

[2.6 交流模块测试 58](#_Toc39859130)

[2.6.1 交流模块白盒测试用例 58](#_Toc39859131)

[2.6.2 交流模块黑盒测试用例 60](#_Toc39859132)

[2.6.3 交流模块测试报告 60](#_Toc39859133)

[2.7 系统集成测试 61](#_Toc39859134)

[2.7.1 集成测试概述 61](#_Toc39859135)

[2.7.2 测试方案 61](#_Toc39859136)

[2.8 系统测试小结 62](#_Toc39859137)

[3 心得体会 63](#_Toc39859138)

# 1 系统实现

## 1.1人员安排

“LSIC社区”大学生学习生活信息交流网站系统实现小组

组长：黄菊

组员：刘畅

侯宝玉

徐建伟

赵肖

## 1.2 系统通用类实现

把经常用到的操作封装在通用类中，需要用到哪种操作的时候直接调用封装好的类就可以实现操作。这样可以实现代码的可重用性，大大节省编写程序的时间。本系统将通用类按功能和模块放入文件夹中，以便形成自己的命名空间，如将数据库通用类按照模块划分到com.mrkj.yql.service.forum.impl文件夹、com.mrkj.yql.service.login.impl文件夹、com.mrkj.yql.service.role.impl文件夹、com.mrkj.yql.service.sys.impl文件夹中。系统部分通用类管理如图5.2所示。

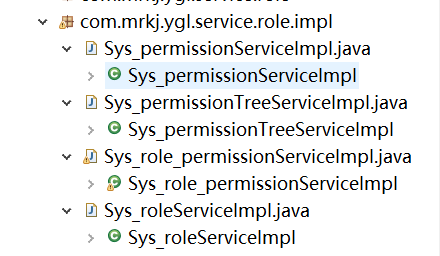


图1.1 系统部分通用类管理

### 1.2.1 数据库通用类实现

数据库通用类按照操作的表的类型划分为四个部分，第一个部分由处理论坛帖子的相关数据表的ForumServiceImpl类组成，第二部分由处理用户注册的相关数据表的Sys\_loginServiceImpl类、UserloginServiceImpl类组成，第三部分由处理用户角色的相关数据表的Sys\_permissionServiceImpl类、Sys\_permissionTreeServiceImpl类组成，第四部分由处理名词字典的DictonaryServiceImpl类组成。这些数据库的通用类是应用程序通过JDBC访问数据库的基础，也是应用程序实现上层业务逻辑的基础。它们利用jdbcTemplate工具类的方法执行编写的SQL语句，提供了查询、删除、更新、插入等数据访问方法，如ForumServiceImpl类的insertMain方法，可以将创建的帖子信息存储到数据库的forum\_main表格中。数据库通用类的数据访问方法如图所示：

表1.1 ForumServiceImpl类中的数据访问方法列表

|  |  |
| --- | --- |
| **函数名** | **函数需要执行的操作** |
| int insertMain (String main\_id,String main\_title,String main\_flag,String main\_type,String main\_content,String main\_creatuser,Integer main\_recommend,String main\_delete,String main\_nickname,Integer main\_zan) | 执行将帖子信息储存到forum\_main表的SQL语句，并且更新论坛的关于帖子的统计信息，返回是否更新成功的信息。 |
| int editJinghua (Integer recommend,String mainId) | 执行更新帖子是精华帖子还是普通帖子信息的SQL语句，返回是否更新成功的信息。 |
| int insertSecond(String mainId,String content,Integer sequence,Integer resequence,String reId,String username,String nickname) | 执行讲回复的帖子信息储存到forum\_second表的SQL语句，并且更新了forum\_info表的info\_reply（回复帖子的个数）、info\_lastuser、info\_lastime信息，更新了base\_userinfo表的base\_huitieshu信息，更新了forum\_minfo表的minfo\_reply的信息，返回是否更新成功的信息。 |
| int getSequence(String mainId) | 执行查询forum\_info表中info\_reply信息的SQL语句，返回获取的信息。 |
| List<Map<String, Object>> getMainPage(String mainType,Integer start,Integer offset) | 查询forum\_main表中某一main\_type类型下的所有的帖子信息，或者main\_recommend = 1，即所有精华帖子信息，返回查询到的帖子列表resultListMapForMainLeftInfo。 |
| List<Map<String,Object>> getMainByTitle(String mainTitle) | 在forum\_main表中根据main\_title进行模糊查询，返回帖子主题符合要求的帖子列表resultListMapForBanzhu。 |
| Map<String, Object> getMinfo(String mainType) | 在forum\_minfo表中查询并返回指定main\_type的版块信息resultMapForMinfo。 |
| List<Map<String,Object>> getBanzhuAndSyslogin (String mainType) | 在forum\_banzhu表中查询指定main\_type的版主信息。 |
| Long getMainCount (String mainType) | 在forum\_main表中查询某一main\_type的帖子的总数，或者查询精华帖子或普通帖子的总数，返回查询的数目。 |
| Map<String,Object> getSecondMain (String mainId) | 在forum\_main表中查询某一main\_id的帖子的信息并返回。 |
| List<Map<String,Object>> getSecond (String mainId,String order,Integer start,Integer offset) | 在forum\_second表中查询主帖的相应的随帖信息并返回。 |
| long getSecondCount (String mainId) | 在forum\_second表中查询主帖相应随帖的数目并返回。 |
| Map<String,Object> getPath2 (String mainType,String mainId) | 在forum\_dictionary表中dict\_key对应的dict\_value，或者在forum\_main表中查询某一类型的主帖信息，将查询到的信息返回。 |
| List<Map<String,Object>> getDictionaryByGroup (String group) | 在forum\_dictionary表中按照dict\_group分类，返回dict\_value、dict\_key信息。 |

表1.2 Sys\_loginServiceImpl类中的数据访问方法列表

|  |  |
| --- | --- |
| **函数名** | **函数需要执行的操作** |
| int deleteByPrimaryKey(Integer loginId) | service访问Mapper映射器接口，执行Sys\_loginMapper中的删除sys\_login表中某一条记录的SQL语句。 |
| int insert(Sys\_login record) | service访问Mapper映射器接口，执行Sys\_loginMapper中的在sys\_login插入一条注册信息的SQL语句 |
| int insertSelective(Sys\_login record) | service访问Mapper映射器接口，执行Sys\_loginMapper中的非完整插入sys\_login表中某一条记录的SQL语句。 |
| Sys\_login selectByPrimaryKey(Integer loginId) | service访问Mapper映射器接口，执行Sys\_loginMapper中的根据login\_id查询sys\_login表中的记录的SQL语句。 |
| int updateByPrimaryKeySelective(Sys\_login record) | service访问Mapper映射器接口，执行Sys\_loginMapper中的可选择的更新sys\_login表中的账号和密码的SQL语句。 |
| int updateByPrimaryKey(Sys\_login record) | service访问Mapper映射器接口，执行Sys\_loginMapper中的更新sys\_login表中的账号和密码的SQL语句。 |
| Sys\_login selectByUsername(String username) | service访问Mapper映射器接口，执行Sys\_loginMapper中的根据username查询sys\_login表中的记录的SQL语句。 |
| Map<String, Long> selectByUsernameCount(String username) | service访问Mapper映射器接口，执行Sys\_loginMapper中的查询sys\_login表中某一username的记录的个数的SQL语句。 |
| int insertSys\_login\_roleSelective(Map<String, Object> parm) | service访问Mapper映射器接口，执行Sys\_loginMapper中的在sys\_login\_role表中可选择的插入记录的SQL语句。 |
| List<Sys\_login> selectPage(Map<String, Object> parm) | service访问Mapper映射器接口，执行Sys\_loginMapper中的查询sys\_login表中的记录的SQL语句。 |
| Map<String, Long> selectCount() | service访问Mapper映射器接口，执行Sys\_loginMapper中的查询sys\_login表中的记录的个数的SQL语句。 |
| int updateSys\_login\_roleSelective(Map<String, Object> parm) | service访问Mapper映射器接口，执行Sys\_loginMapper中的可选择的更新sys\_login\_role表中的记录的SQL语句。 |
| List<Map<String, Object>> selectLoginRole(Map<String,Object> parm) | 在sys\_login表格中查询相应的记录。 |

表1.3 UserloginServiceImpl类中的数据访问方法列表

|  |  |
| --- | --- |
| **函数名** | **函数需要执行的操作** |
| int deleteByPrimaryKey(UserloginKey key) | service访问Mapper映射器接口，执行UserloginMapper中的删除user\_login表中某一条记录的SQL语句。 |
| int insert(Userlogin record) | service访问Mapper映射器接口，执行UserloginMapper中的在user\_login插入一条注册信息的SQL语句 |
| int insertSelective(Userlogin record) | service访问Mapper映射器接口，执行UserloginMapper中的在user\_login表中可选择的插入一条注册信息的SQL语句 |
| Userlogin selectByPrimaryKey(UserloginKey key) | service访问Mapper映射器接口，执行UserloginMapper中的查询user\_login表中某一条记录的SQL语句。 |
| int updateByPrimaryKeySelective(Userlogin record) | service访问Mapper映射器接口，执行UserloginMapper中的可选择更新user\_login表中的记录的SQL语句。 |
| int updateByPrimaryKey(Userlogin record) | service访问Mapper映射器接口，执行UserloginMapper中的更新user\_login表中的记录的SQL语句。 |
| Userlogin selectByUsername(String username) | service访问Mapper映射器接口，执行UserloginMapper中的查询user\_login表中特定username的记录的SQL语句。 |
| List<Userlogin> selectPage(Map<String, Object> parm) | service访问Mapper映射器接口，执行UserloginMapper中的查询user\_login表中某一页记录的SQL语句。 |
| Userlogin selectByPrimaryKeyUsername(UserloginKey key) | service访问Mapper映射器接口，执行UserloginMapper中的根据id和username查询user\_login表中的记录的SQL语句。 |
| Map<String, Long> selectCount(Map<String, Object> parmMap) | service访问Mapper映射器接口，执行UserloginMapper中的查询user\_login表中记录的数目SQL语句。 |

表1.4 Sys\_permissionServiceImpl类中的数据访问方法列表

|  |  |
| --- | --- |
| **函数名** | **函数需要执行的操作** |
| int deleteByPrimaryKey(Integer permissionId) | service访问Mapper映射器接口，执行Sys\_Sys\_permissionServiceImpl中的删除sys\_permission表中某一条记录的SQL语句。 |
| int insert(Sys\_permission record) | service访问Mapper映射器接口，执行Sys\_permissionServiceImpl中的在sys\_permission插入一条注册信息的SQL语句 |
| int insertSelective(Sys\_permission record) | service访问Mapper映射器接口，执行Sys\_permissionServiceImpl中的在sys\_permission表中可选择的插入一条信息的SQL语句 |
| Sys\_permission selectByPrimaryKey(Integer permissionId) | service访问Mapper映射器接口，执行Sys\_permissionServiceImpl中的根据permission\_id查询sys\_permission表中记录的SQL语句。 |
| int updateByPrimaryKeySelective(Sys\_permission record) | service访问Mapper映射器接口，执行Sys\_permissionServiceImpl中的可选择更新sys\_permission表中的记录的SQL语句。 |
| int updateByPrimaryKey(Sys\_permission record) | service访问Mapper映射器接口，执行Sys\_permissionServiceImpl中的更新sys\_permission表中的记录的SQL语句。 |

表1.5 Sys\_permissionTreeServiceImpl类中的数据访问方法列表

|  |  |
| --- | --- |
| **函数名** | **函数需要执行的操作** |
| List<Sys\_permissionTree> selectAll() | service访问Mapper映射器接口，执行Sys\_permissionTreeMapper中的查询sys\_permission表中的所有记录的SQL语句。 |
| int deleteByPrimaryKey(Integer id) | service访问Mapper映射器接口，执行Sys\_permissionTreeMapper中的删除sys\_permission表中的信息的SQL语句 |

表1.6 Sys\_role\_permissionServiceImpl类中的数据访问方法列表

|  |  |
| --- | --- |
| **函数名** | **函数需要执行的操作** |
| int deleteByPrimaryKey(Integer id) | service访问Mapper映射器接口，执行Sys\_role\_permissionMapper中的删除sys\_role\_permission表中某一条记录的SQL语句。 |
| int insert(Sys\_role\_permission record) | service访问Mapper映射器接口，执行Sys\_role\_permissionMapper中的在sys\_role\_permission插入一条注册信息的SQL语句 |
| int insertSelective(Sys\_role\_permission record) | service访问Mapper映射器接口，执行Sys\_role\_permissionMapper中的在sys\_role\_permission表中可选择的插入一条信息的SQL语句 |
| Sys\_role\_permission selectByPrimaryKey(Integer id) | service访问Mapper映射器接口，执行Sys\_role\_permissionMapper中的根据id查询sys\_role\_permission表中记录的SQL语句。 |
| int updateByPrimaryKeySelective(Sys\_role\_permission record) | service访问Mapper映射器接口，执行Sys\_role\_permissionMapper中的可选择更新sys\_role\_permission表中的记录的SQL语句。 |
| int updateByPrimaryKey(Sys\_role\_permission record) | service访问Mapper映射器接口，执行Sys\_role\_permissionMapper中的更新sys\_role\_permission表中的记录的SQL语句。 |
| int deleteByRoleName(String roleName) | service访问Mapper映射器接口，执行Sys\_role\_permissionMapper中的根据role\_name删除sys\_role\_permission表中的记录的SQL语句。 |
| int updateRolPer(String roleName, Integer roleId, String[] splitParm) | service访问Mapper映射器接口，执行Sys\_role\_permissionMapper中的可选择的更新sys\_role\_permission表中的记录的SQL语句。 |
| List<Sys\_role\_permission> selectByRoleName(String roleName) | service访问Mapper映射器接口，执行Sys\_role\_permissionMapper中的根据role\_name查询sys\_role\_permission表中的记录的SQL语句。 |

表1.7 Sys\_roleServiceImpl类中的数据访问方法列表

|  |  |
| --- | --- |
| **函数名** | **函数需要执行的操作** |
| int deleteByPrimaryKey(Integer roleId) | service访问Mapper映射器接口，执行Sys\_roleMapper中的删除sys\_role表中某一条记录的SQL语句。 |
| int insert(Sys\_role record) | service访问Mapper映射器接口，执行Sys\_roleMapper中的在sys\_role插入一条注册信息的SQL语句 |
| int insertSelective(Sys\_role record) | service访问Mapper映射器接口，执行Sys\_roleMapper中的在sys\_role表中可选择的插入一条信息的SQL语句 |
| Sys\_role selectByPrimaryKey(Integer roleId) | service访问Mapper映射器接口，执行Sys\_roleMapper中的根据role\_id查询sys\_role表中记录的SQL语句。 |
| int updateByPrimaryKeySelective(Sys\_role record) | service访问Mapper映射器接口，执行Sys\_roleMapper中的可选择更新sys\_role表中的记录的SQL语句。 |
| int updateByPrimaryKey(Sys\_role record) | service访问Mapper映射器接口，执行Sys\_roleMapper中的更新sys\_role表中的记录的SQL语句。 |
| List<Sys\_roleTree> selectTreeAll() | service访问Mapper映射器接口，执行Sys\_roleMapper中的查询sys\_role表中所有的记录的SQL语句。 |

表1.8 DictonaryServiceImpl类中的数据访问方法列表

|  |  |
| --- | --- |
| **函数名** | **函数需要执行的操作** |
| int deleteByPrimaryKey(Integer dictId) | service访问Mapper映射器接口，执行DictionaryMapper中的删除forum\_dictionary表中某一条记录的SQL语句。 |
| int insert(Dictionary record) | service访问Mapper映射器接口，执行DictionaryMapper中的在forum\_dictionary插入一条注册信息的SQL语句 |
| int insertSelective(Dictionary record) | service访问Mapper映射器接口，执行DictionaryMapper中的在forum\_dictionary表中可选择的插入一条信息的SQL语句 |
| Dictionary selectByPrimaryKey(Integer dictId) | service访问Mapper映射器接口，执行DictionaryMapper中的根据forum\_dictionary查询sys\_role表中记录的SQL语句。 |
| int updateByPrimaryKeySelective(Dictionary record) | service访问Mapper映射器接口，执行DictionaryMapper中的可选择更新forum\_dictionary表中的记录的SQL语句。 |
| int updateByPrimaryKey(Dictionary record) | service访问Mapper映射器接口，执行DictionaryMapper中的更新forum\_dictionary表中的记录的SQL语句。 |
| List<Dictionary> selectByGroup(String dictGroup) | service访问Mapper映射器接口，执行DictionaryMapper中的根据dict\_group查询forum\_dictionary表中的记录的SQL语句。 |

### 1.2.2 其它通用类实现

（1）基本对象操作类CCommon：把对基本类型的操作封装在CCommon类里面。CCommon类的主要功能是打印异常信息、比较字符串是否相同、忽略大小写比较字符串是否相同、比较两个字符串、忽略大小写比较两个字符串、检查字符串中是否包含另一字符串、忽略大小写、字符串转换整数、字符串转换字节、字符串是否等于null或者Empty、根据指定的字符串将字符串分割为字符串数组、根据指定的字符串将字符串分割为整数数组、指定的类型是否是包装类、指定的类型是否是包装类或者基本类型、比较两个对象是否相等、转换对象为Long对象、转换对象为Long对象、将一组整形数字组合成一个以指定分隔符分割成的字符串、将一组整形数字组合成一个以指定分隔符分割成的字符串、获取目录名称，如果目录以“/”或者“\”结尾，去除掉结尾字符、创建文件等操作。

（2）字符串类CStringUtil：把对字符串的操作封装在CStringUtil类里面。CstringUtil的主要功能是格式化字符串、首字母转大写其他不变、循环拼接字符串、字符串组装等操作。

（3）数据格式化类DateUtil：把对数据格式化的操作放在DateUtil类中。DateUtil的主要功能是获取YYYY格式、获取YYYY-MM-DD格式、获取YYYYMMDD格式、获取YYYY-MM-DD HH:mm:ss格式、格式化日期、校验日期是否合法等操作。

## 1.3 主页面模块实现

### 1.3.1 主页面模块实现简介

用户打开网站后，首先接触的是网站的主页面，主页面包含“LIFE 生活专区”、“STUDY 学习专区”、“INFORMATION 资源专区”、“COMMUNICATION 交流专区”四个专区，每个专区下有相应子模块展示，可以点击进入。另外主页还包含精华帖、登录选择，用户可以选择这些功能。在此页面，无论用户是否登录，所看到的页面都是相同的。此模块的详细模块图如下：



图1.1 主页面模块详细图

其中主要包括五个子模块，每个子模块还有各自的功能。

LIFE生活服务专区：包含代取快递、二手交易、租住房屋、组团拼单/拼车/出游、兼职版块，用户可直接点击版块名浏览相应版块中的帖子。主页中，在每个专区的各个版块下，都会展示相应版块中的部分精华帖名，用户可直接点击帖名查看该帖内容与回帖。在此专区为用户提供生活服务，用户可以在相应的版块发帖寻找代取快递者，或查看二手商品，或查看租房信息等。

STUDY学习专区：包含考研/保研版块，用户可直接点击版块名浏览相应版块中的帖子。主页中，在每个专区的各个版块下，都会展示相应版块中的部分精华帖名，用户可直接点击帖名查看该帖内容与回帖。此专区为用户提供考研/保研资料的分享交流平台，用户可在改专区的版块下，查看其他用户发布的关于考研/保研帖，或发布有关考研/保研的帖子。

INFORMATION专区：包含资源版块，用户可直接点击版块名浏览相应版块中的帖子。主页中，在每个专区的各个版块下，都会展示相应版块中的部分精华帖名，用户可直接点击帖名查看该帖内容与回帖。该专区为用户提供资源上传与下载平台，用户在发帖时可以上传相关资源共其他用户下载。

精华子页面：在主页面中以一个按钮的形式存在，在此子页面中，呈现此网站各个版块的所有精华帖，用户可以集中进行查看。

登录与注册页面：在主页面中以一个按钮的形式存在，用户只有登录才能在子页面中发帖，否则只能浏览帖子。

该模块用到的数据库表有forum\_dictionary表（dict\_id, dict\_group, dict\_key, dict\_value, dict\_parent, fict\_order, dict\_isleaf），forum\_minfo表（minfo\_id, main\_type, minfo\_reply, minfo\_banzhu, minfo\_creatime, minfo\_creatuser, minfo\_order, minfo\_see,minfo\_photo, minfo\_title, minfo\_count），my\_main表（main\_id, main\_title, main\_content, main\_creatime, main\_creatuser, main\_commend）。

### 1.3.2 主页面模块相关类实现

1.论坛组件类ForumComponent：把对论坛主页组件的操作封装在ForumComponent类中，MoocFragment类的成员变量和方法如下：

@Component(value="forumComponent")

**public** **class** ForumComponent {

@Value("#{setting['forumList']}")

**private** String forumList;

**public** String getForumList() {

**return** forumList;

}

**public** **void** setForumList(String forumList) {

**this**.forumList = forumList;

}

}

2.论坛专区的子版块类Dictionary：把对论坛子版块的操作封装在Dictionary类中，与数据库中forum\_dictionary表对应，实现后端与数据库此部分的操作。此类的成员变量与方法如下：

**public** **class** Dictionary {

**private** Integer dictId;

**private** String dictGroup;

**private** String dictKey;

**private** String dictValue;

**private** String dictParent;

**private** Integer dictOrder;

**private** String dictIsleaf;

**public** Integer getDictId() {

**return** dictId;

}

**public** **void** setDictId(Integer dictId)

**public** String getDictGroup()

**public** **void** setDictGroup(String dictGroup)

**public** String getDictKey()

**public** **void** setDictKey(String dictKey)

**public** String getDictValue()

**public** **void** setDictValue(String dictValue)

**public** String getDictParent()

**public** **void** setDictParent(String dictParent)

**public** Integer getDictOrder()

**public** **void** setDictOrder(Integer dictOrder) {

**public** String getDictIsleaf()

**public** **void** setDictIsleaf(String dictIsleaf)

}

3.主页控制器类IndexController:实现主页的导航与主页的初始化，包括主页中四个专区的初始化、每个专区中各个版块的初始化和主页中各个版块里展示的贴名。此类的成员变量与方法如下：  
@Controller

**public** **class** IndexController {

@Resource

DictionaryService dictS;

@Resource

ForumServiceImpl forS;

@Resource

ForumComponent fc;

@RequestMapping(value={"/index","/index.do","/index.html","index.action"})

**public** ModelAndView initIndex (){

ModelAndView mav = **new** ModelAndView("index");

String forumList = fc.getForumList();

String[] forums = forumList.split(",");

List<Map<String,List<forumMain>>> top5s = **new** ArrayList<>();

**for** (String tempStr : forums ){

List<Dictionary> forumTypes = dictS.selectByGroup(tempStr);

mav.addObject(tempStr, forumTypes);

**for** (Dictionary tempEntity : forumTypes){

String typeKey = tempEntity.getDictKey();

List<forumMain> top5 = forS.getRecommendTop5(typeKey);

**if** (top5!=**null**&&top5.size()>0){

Map<String,List<forumMain>> top5Map = **new** HashMap<>();

top5Map.put(typeKey, top5);

top5s.add(top5Map);

}

}

}

mav.addObject("top5s", top5s);

**return** mav;

}

}

4.主页动态网页文件index.jsp:实现网站主页动态初始化。

5.各个跳转的子页面，如mainContent.jsp、secondContent等。

### 1.3.3 主页面模块文件及跳转关系

示例：

（1）模块涉及代码文件列表如表2.1所示。

表1.9 主页面模块涉及代码文件列表

|  |  |  |
| --- | --- | --- |
| **文件名** | **文件路径** | **文件说明** |
| index.jsp | F:\javaproject\mrbbs\WebContent\WEB-INF\view | 网站动态主页面 |
| mainContent.jsp | F:\javaproject\mrbbs\WebContent\WEB-INF\view | 网站动态子版块页面 |
| secondContent.jsp | F:\javaproject\mrbbs\WebContent\WEB-INF\view | 网站动态查看与回复帖子页面 |
| Jinghua.jps | F:\javaproject\mrbbs\WebContent\WEB-INF\view | 网站动态精华帖集中查看页面 |
| Login.jsp | F:\javaproject\mrbbs\WebContent\WEB-INF | 网站注册与登录页面 |

（2）模块涉及页面跳转关系图，如图1.2所示。



图1.2主页面模块涉及界面跳转关系图

### 1.3.4主页面模块程序流程

主页面模块实现程序流程如图1.3所示。



图1.3 主页面模块程序流程图

### 1.3.5 主页面模块实现界面

主页面模块实现界面如图1.4-1.5所示。

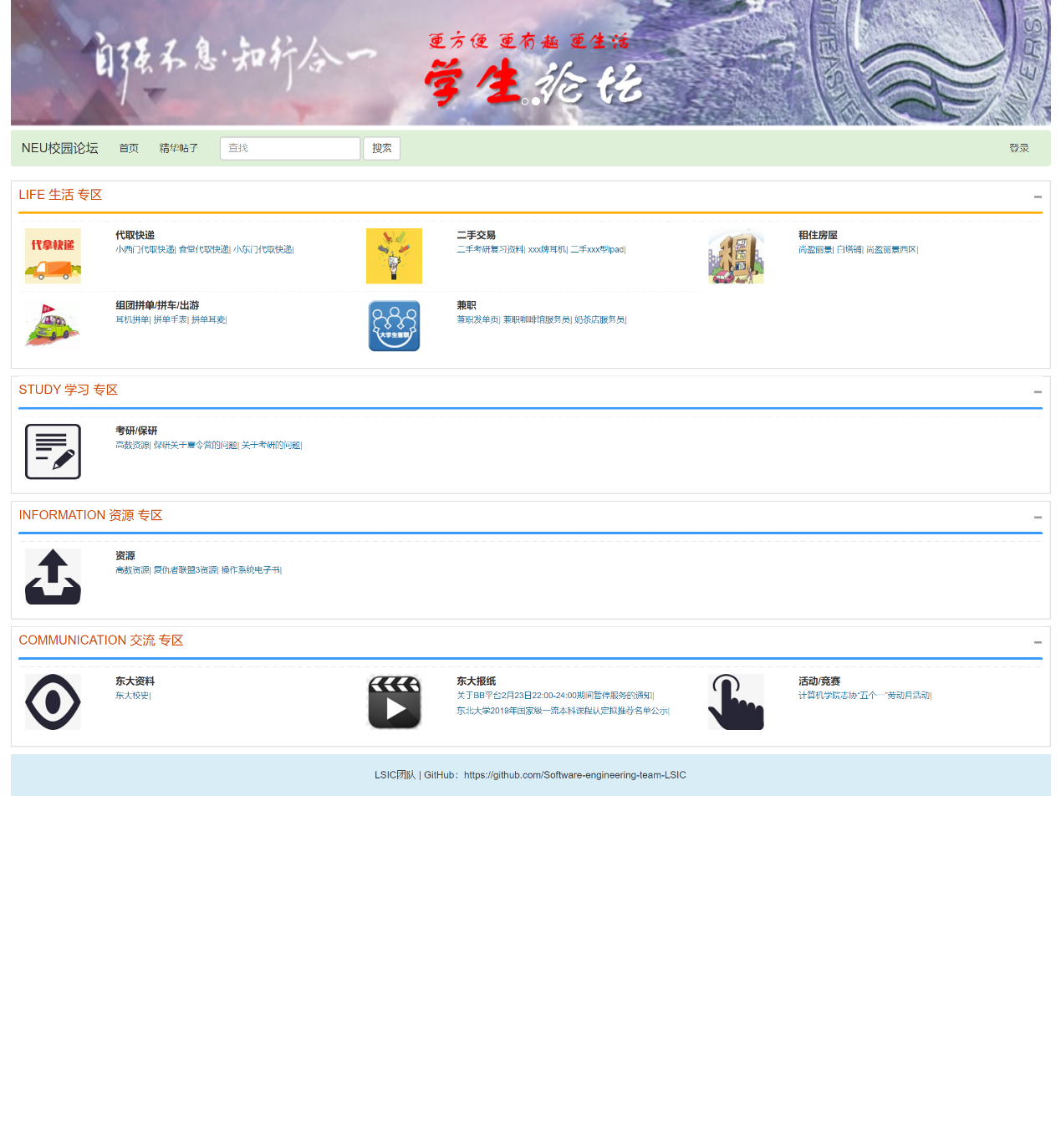


图1.4 主页面模块实现界面

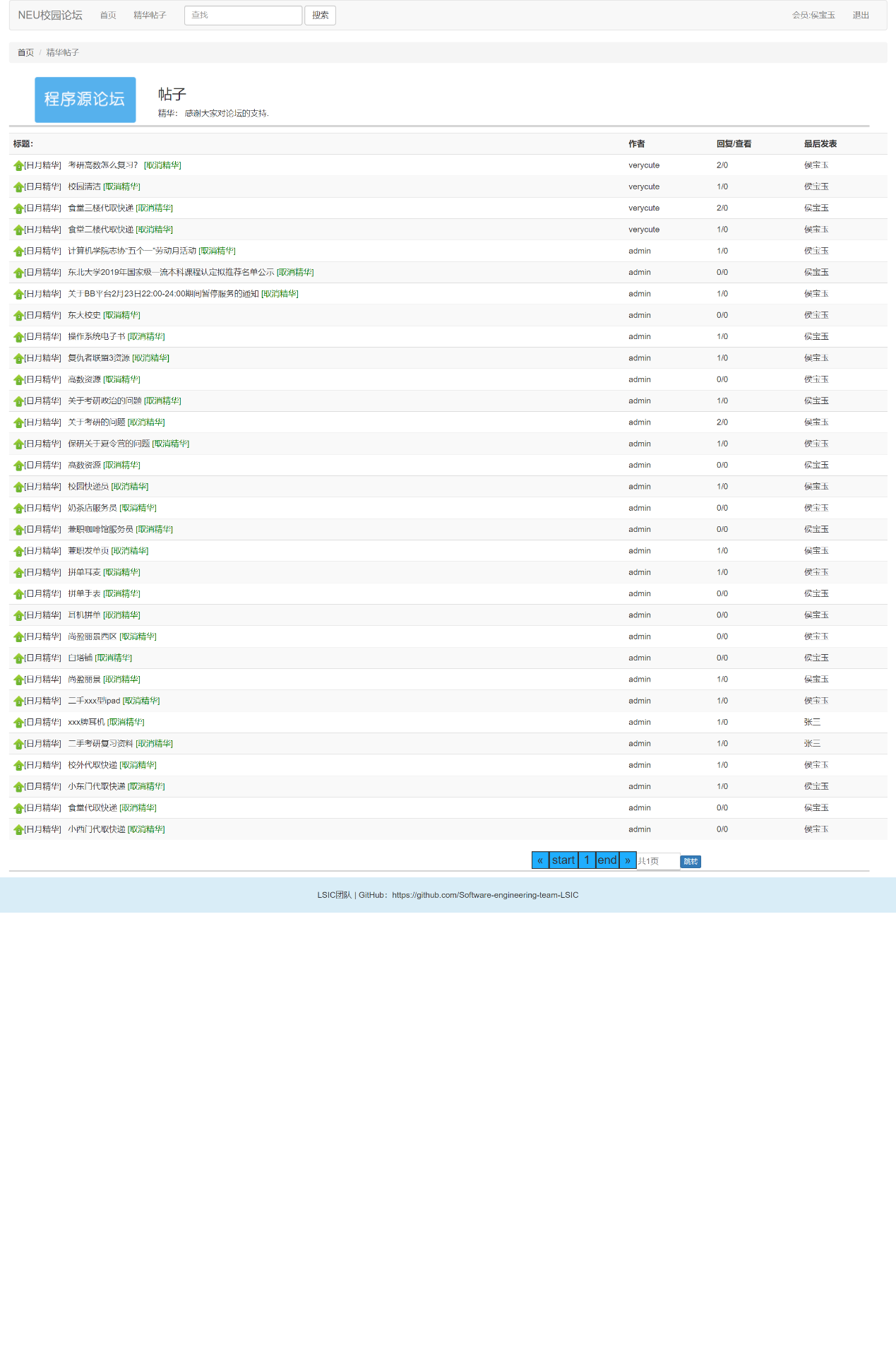


图1.5精华帖子实现界面

## 1.4 分类专区模块实现

### 1.4.1 分类专区模块实现简介

本团队的分类专区包括生活专区、学习专区、资源专区三个大的功能区域。其中生活专区内包括代取快递、组团拼单/拼车/出游、二手交易、兼职细分功能区。学习专区包括考研保研细分功能区。资源专区是供学生下载和上传实现资源共享的区域。交流专区包括东大资料、东大报纸、活动/竞赛细分功能区。学生可以通过首页的这些功能区标志进入对应的功能区查看精华帖子内容，回复帖子，发布帖子。此功能区涉及的数据库有：forum\_banzhu、forum\_dictionary、 forum\_info、forum\_main、forum\_minfo、forum\_second。

### 1.4.2 分类专区模块相关类实现

用户登录所涉及到的类为MainPageController，该类中的属性MainPageService mps用于向数据库中传递信息。该类中的方法有：

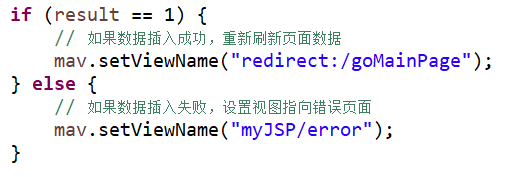
方法一：

public ModelAndView saveUeditor(HttpServletRequest request, String content, String mainTitle)

获取客户端IP地址作为发帖人



根据result值判断是否成功向数据库中插入了内容，如果没有调转到错误页面：



方法二：

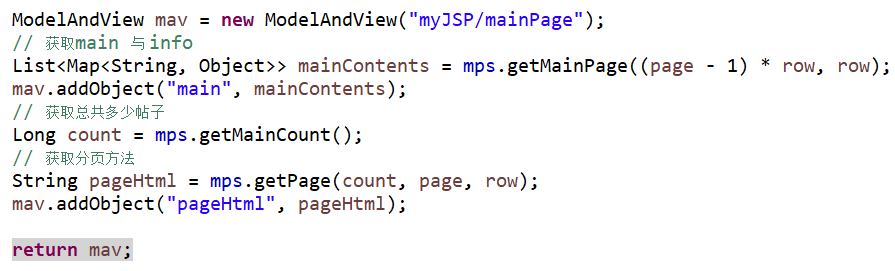
public ModelAndView goMainPage(HttpServletRequest request,

@RequestParam(name = "page", defaultValue = "1") Integer page,

@RequestParam(name = "row", defaultValue = "40") Integer row)

该方法用于初始化论坛主页面

从数据库中获取本功能区内的帖子内容和帖子数量进行显示：



涉及到的第二个类为MainPageService。该类用于与数据库建立链接，从数据库中读取数据或者向数据库中写入数据。

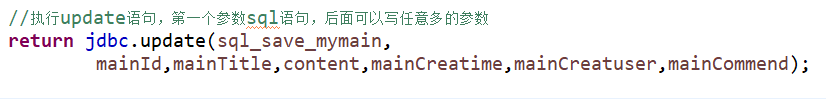
方法一：

public int saveMainContent(String content,String mainTitle,String mainCreatuser)该方法从前端网页获取用户的发帖信息，并将获取到的信息插入数据库中。

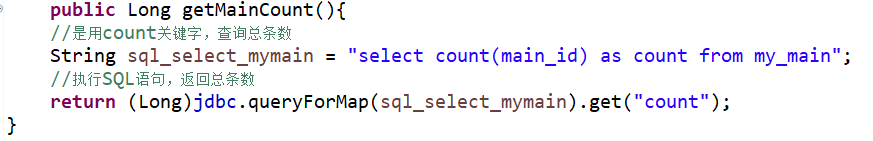
定义插入SQL语句：



使用jdbc向数据库中插入信息：



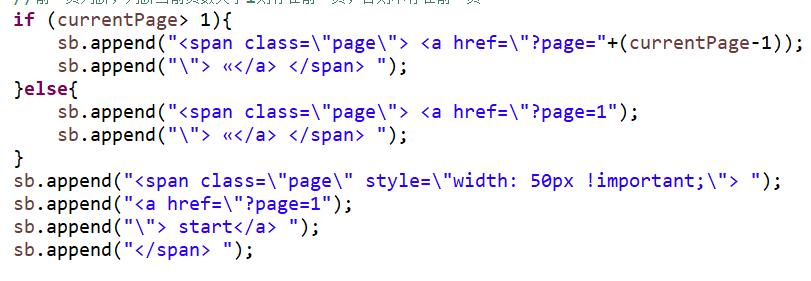
方法二：

public Long getMainCount()方法用于在数据库中查询帖子数量。

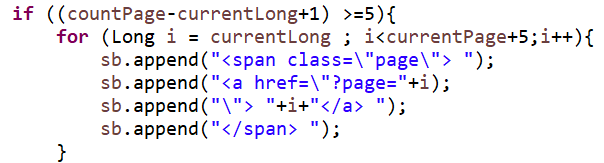
方法三：

public String getPage (Long count,Integer currentPage,Integer offset)方法用于设置显示页面导航格式。

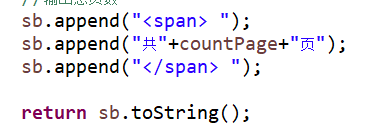
前一页判断，判断当前页数大于1则存在前一页，否则不存在前一页：



中间页数导航，中间最多显示5页：



输出总页数：



### 1.4.3 分类专区模块文件及跳转关系

（1）模块涉及代码文件列表如表1.10所示。

表1.10 用户注册模块涉及代码文件列表

|  |  |  |
| --- | --- | --- |
| **文件名** | **文件路径** | **文件说明** |
| index.jsp | mrbbs\WebContent\WEB-INF\view\index.jsp | 网站首页，显示各个功能分区内容，并可以通过该页跳转到其他页面 |
| mainContent.jsp | mrbbs\WebContent\WEB-INF\view\mainContent.jsp | 设计显示各个分区的具体内容 |
| MainPageController.java | mrbbs\Java Resources\myresource\MainPageController.java | 控制从前端接受用户提交的内容，并传给server |
| MainPageService.java | mrbbs\Java Resources\myresource\MainPageService.java | 与数据库连接把前端接受到的用户输入信息存入数据库，从数据库中读取内容返回前端显示 |

（2）模块涉及页面跳转关系图，如图1.6所示。

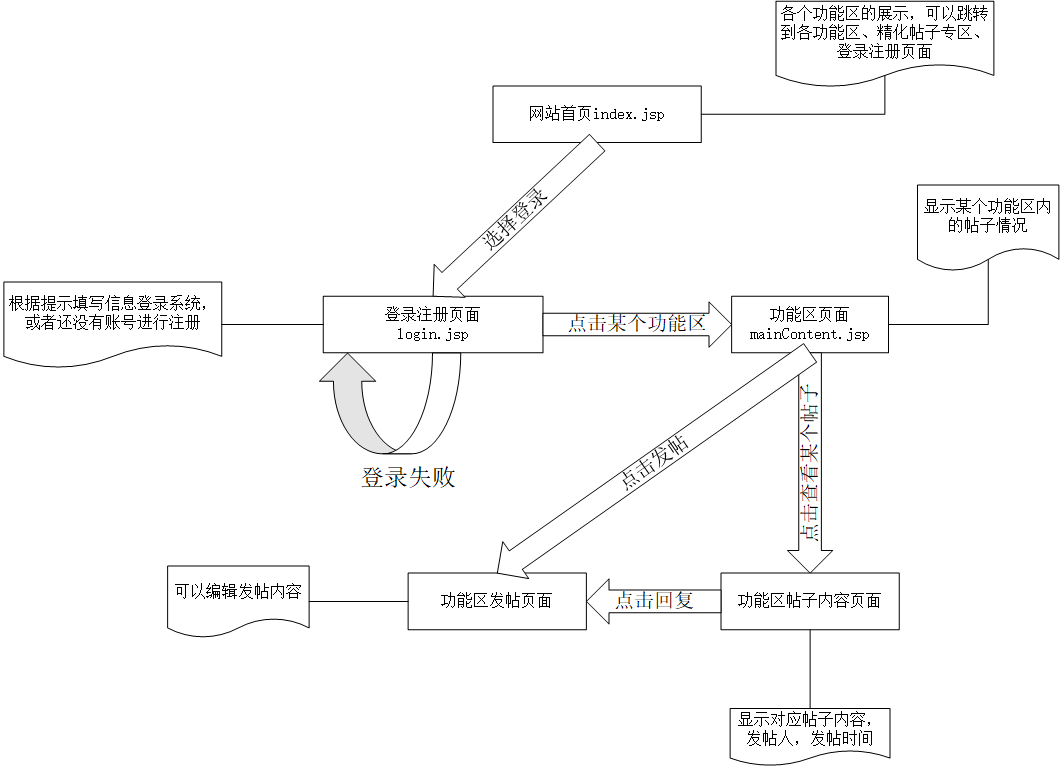


图1.6 分类专区模块涉及界面跳转关系图

### 1.4.4 分类专区模块程序流程

分类专区模块实现程序流程如图2.2所示。用户在首页可以看到本网站所有的功能区包括生活专区、学习专区、资源专区三个大的功能区域。其中生活专区内包括代取快递、组团拼单/拼车/出游、二手交易、兼职细分功能区。学习专区包括考研保研细分功能区。资源专区是供学生下载和上传实现资源共享的区域。交流专区包括东大资料、东大报纸、活动/竞赛细分功能区。如果用户没有登录可以通过单击各个功能区进入功能区帖子展示页面，点击具体的帖子可以看到帖子的内容，发帖人，发帖时间等信息；如果用户登录进入系统可以进行更多的活动比如发帖子，设置精华帖子，回复他人帖子等活动。

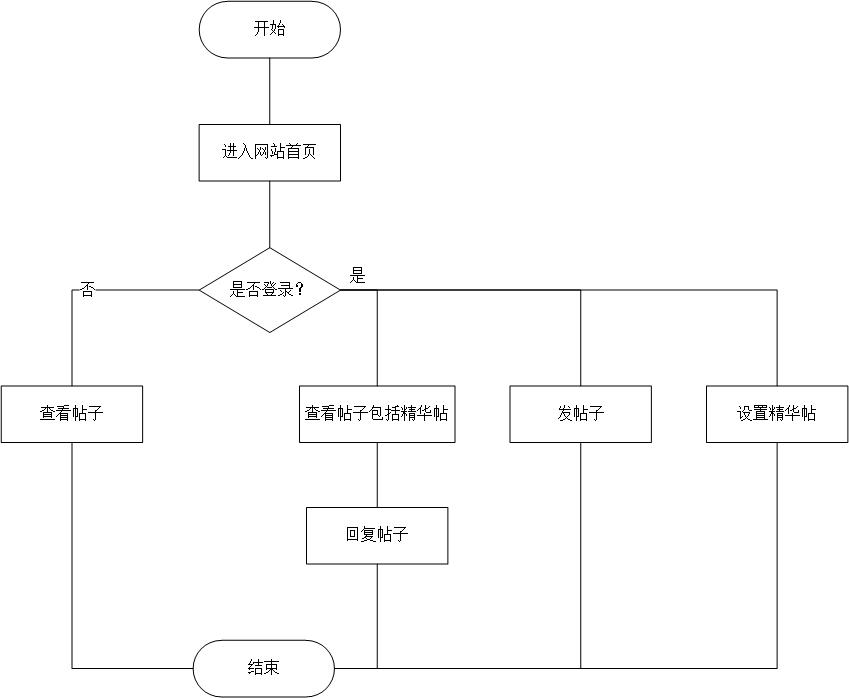


图1.7分类专区模块程序流程图

### 1.4.5 分类专区模块实现界面

分类专区模块实现界面如图1.8-1.17所示。

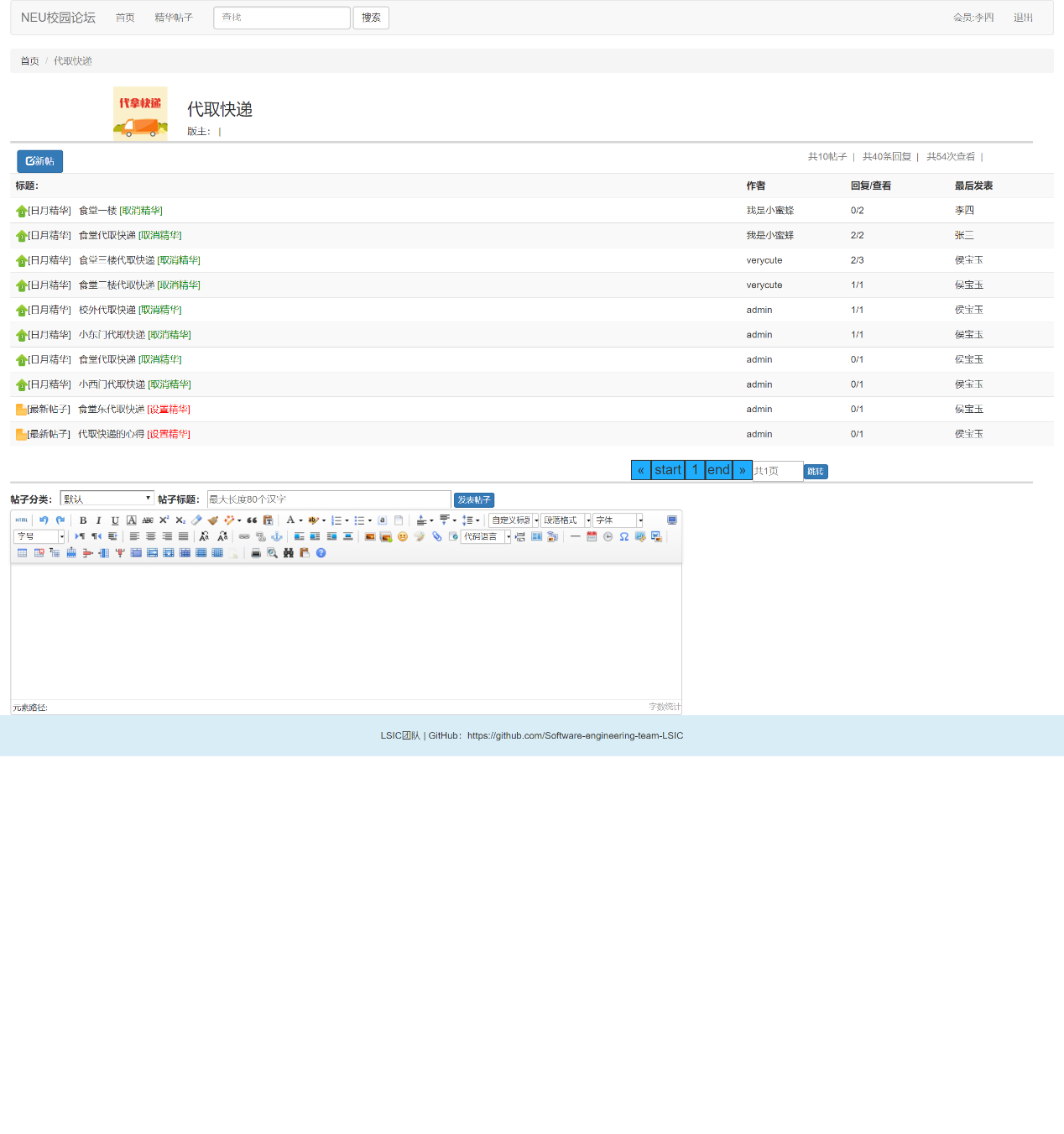


图1.8代取快递功能区实现界面

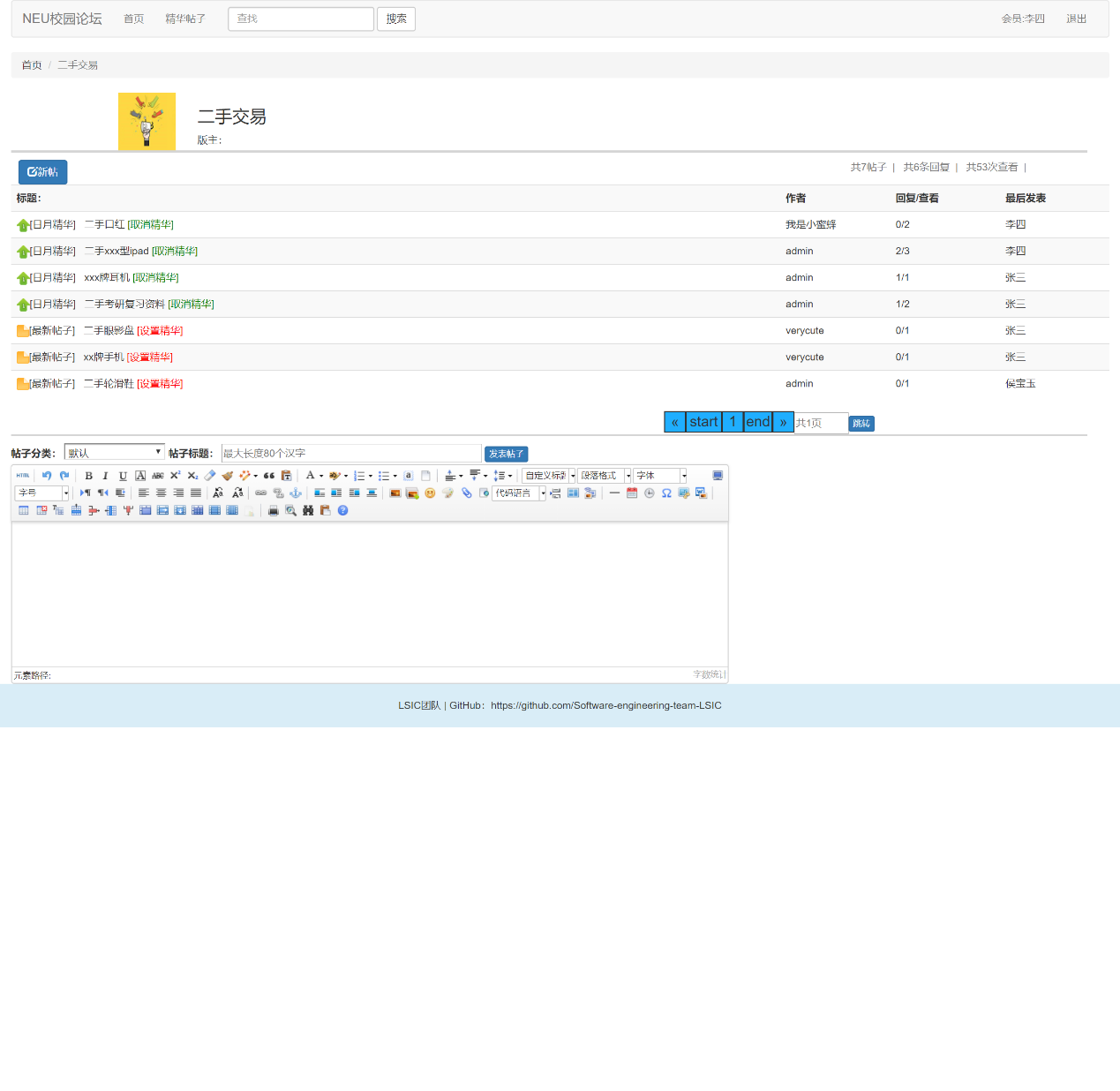


图1.9二手交易功能区实现界面

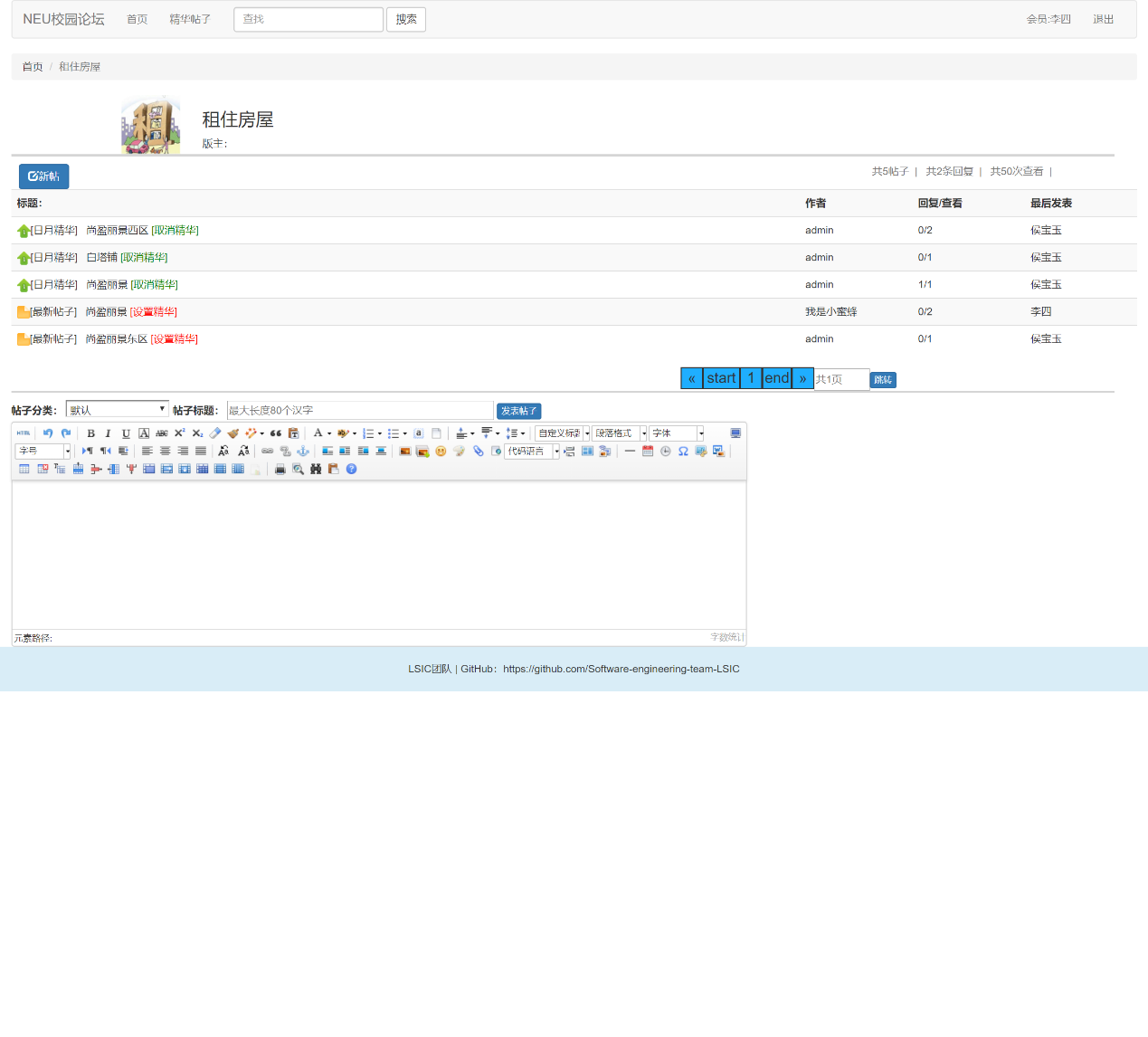


图1.10租住房屋功能区实现界面

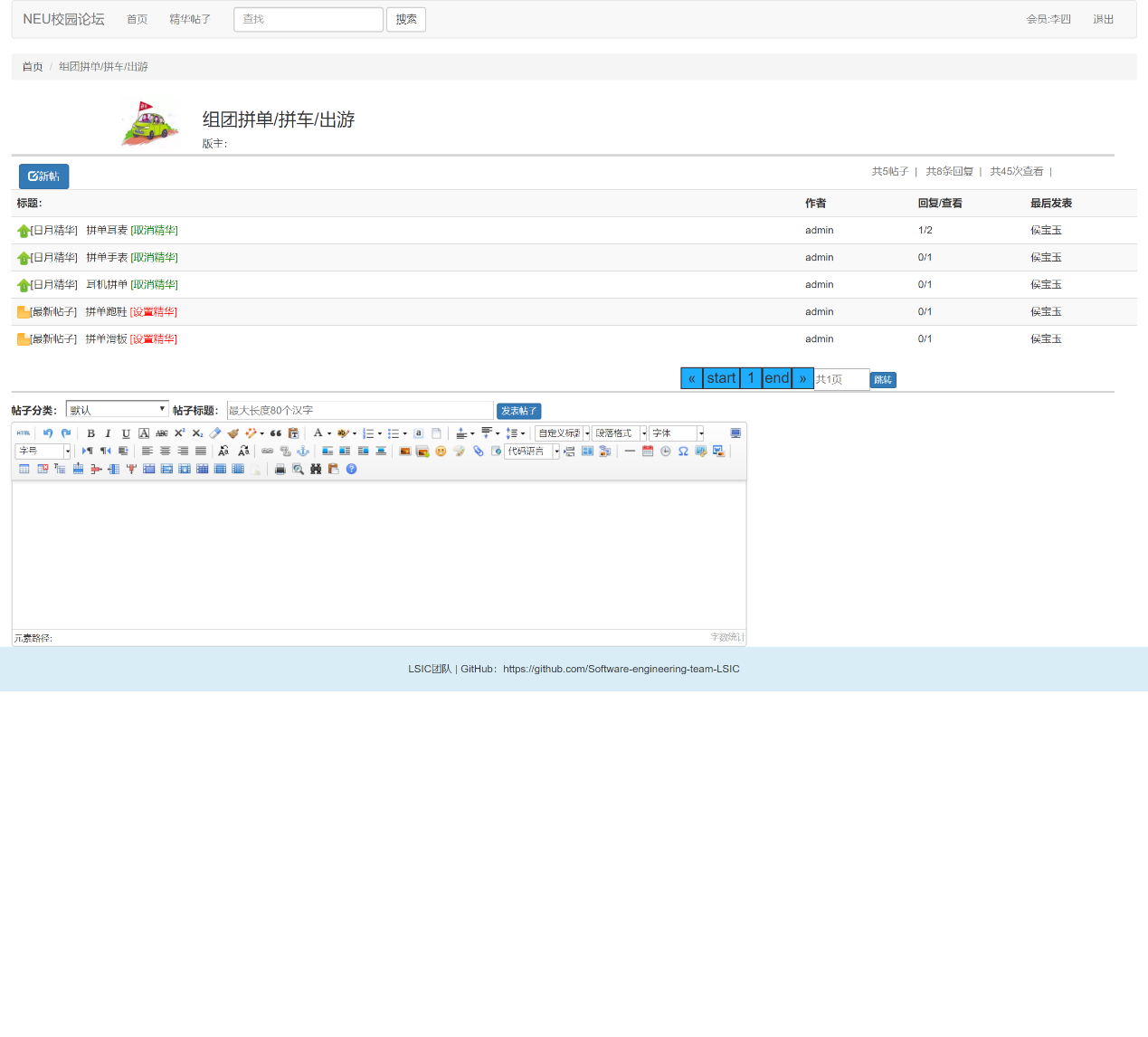


图1.11组团拼车/拼单/出游功能区实现界面

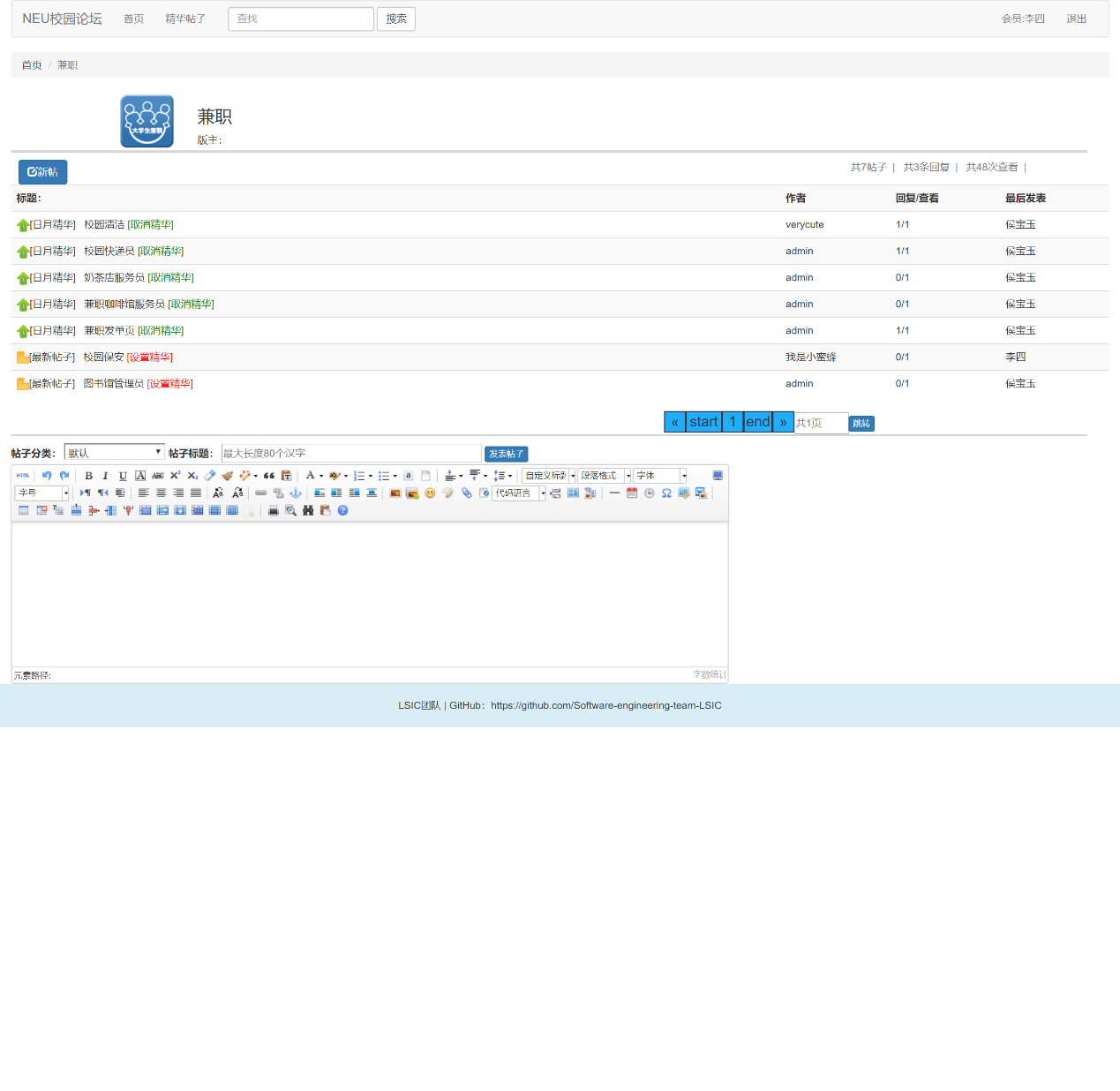


图1.12兼职功能区实现界面

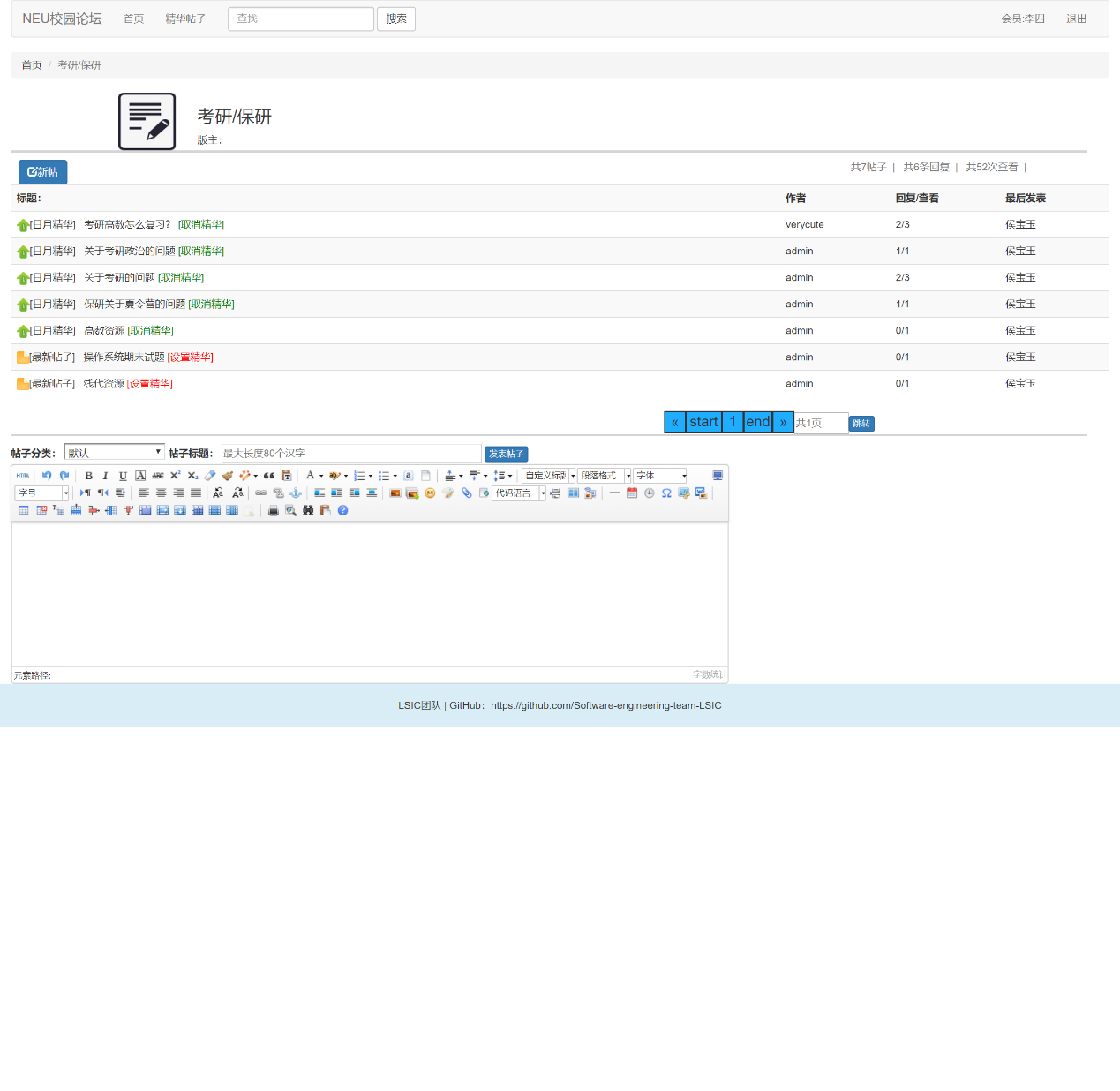


图1.13考研/保研功能区实现界面

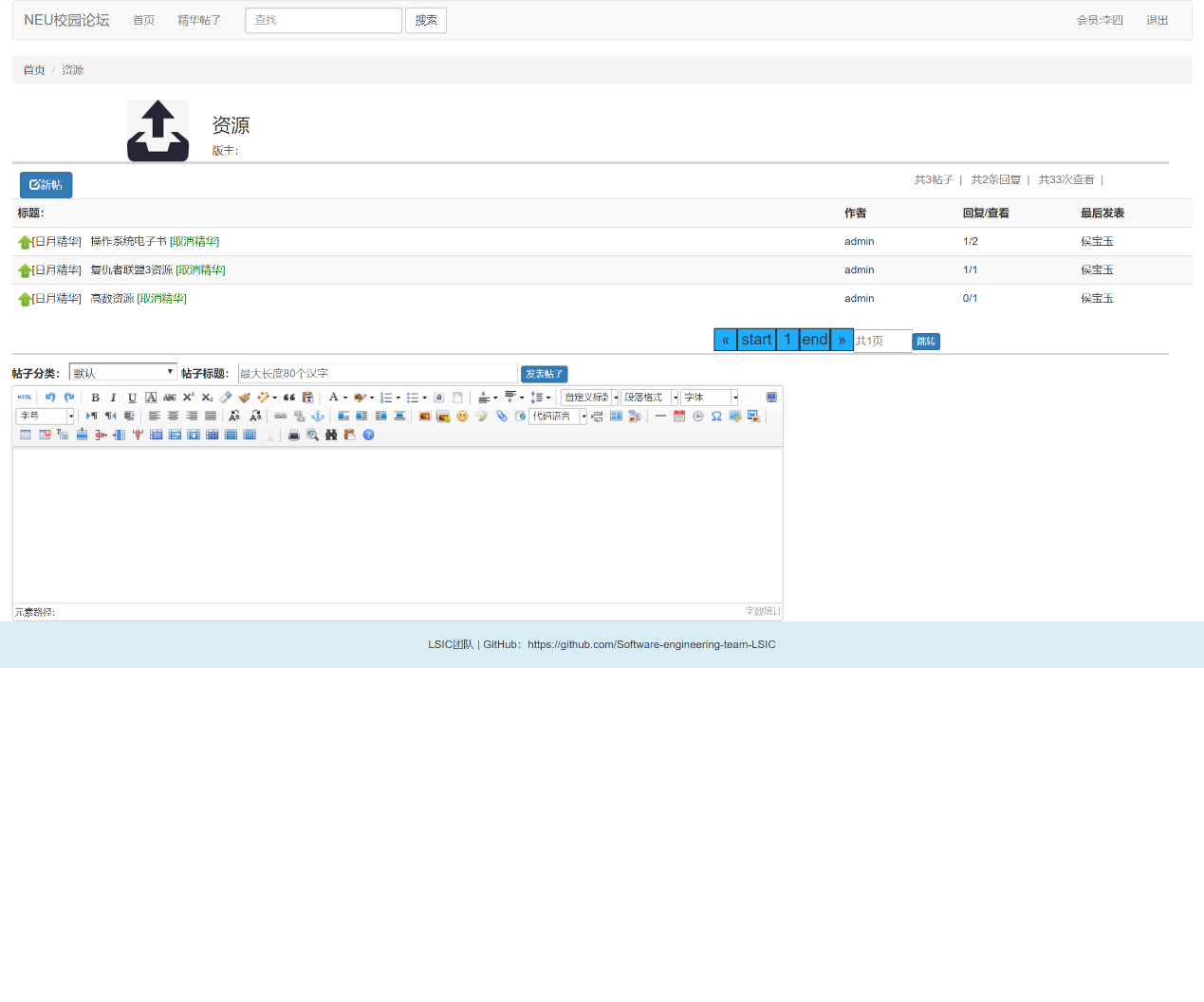


图1.14资源功能区实现界面

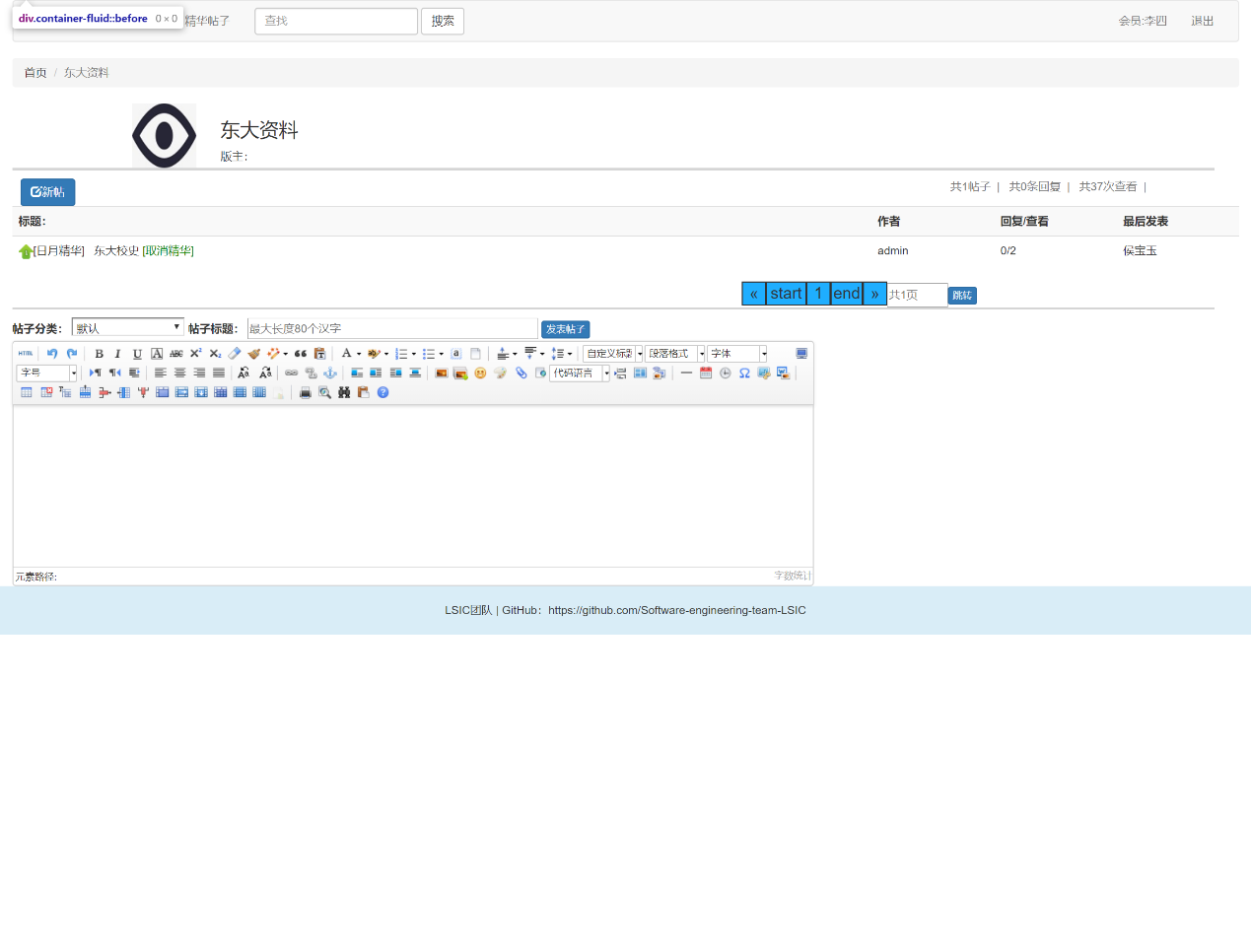


图1.15东大资料功能区实现界面

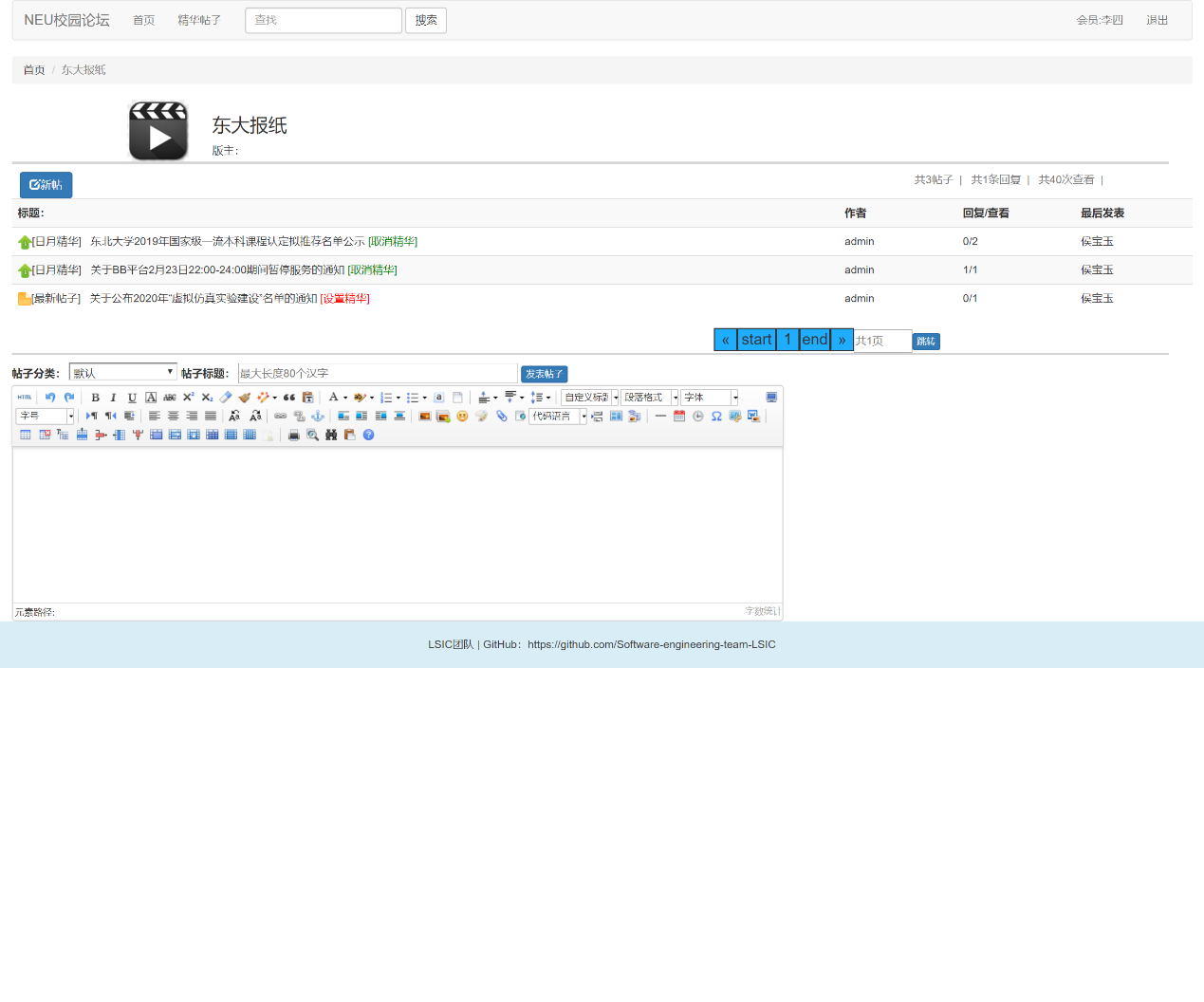


图1.16东大报纸功能区实现界面

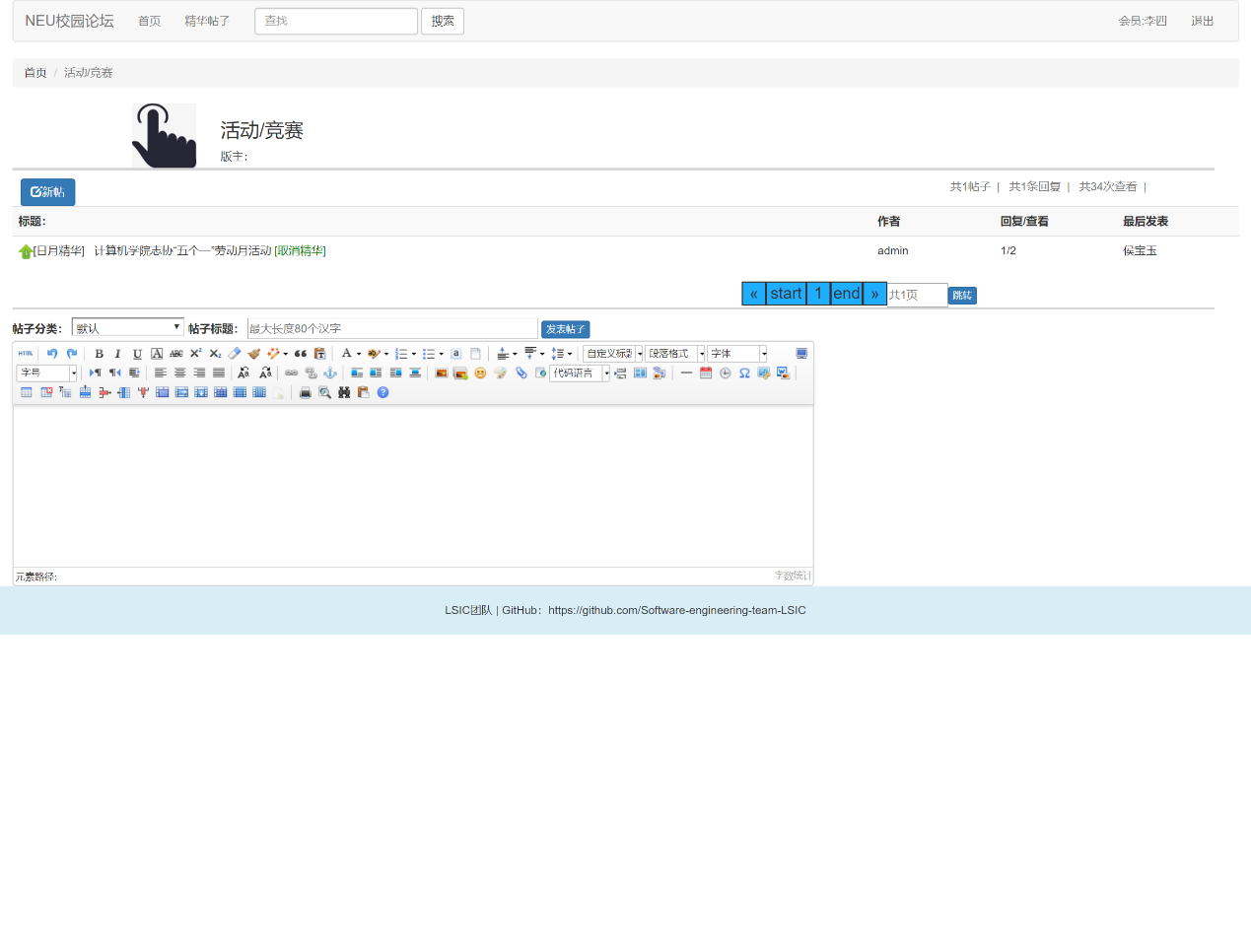


图1.17活动/竞赛功能区实现界面

## 1.5 查看与回复帖子模块实现

### 1.5.1 查看与回复帖子模块实现简介

用户在主页面或者某版块页面中，点击某一帖子，将进入此页面。此页面包含楼主发的帖子、发帖内容、发帖时间、楼主信息、回帖内容、回帖人信息等。用户在登录后，可以回复当前所在贴，回复内容可以是文字、图片、链接等。

该模块用到的数据库表有base\_userinfo表（base\_id,base\_name,base\_sex,base\_age,base\_fatieshu, base\_huitieshu, base\_photo, base\_shenfenzhen,username），forum\_banzhu表（banzhu\_id, main\_type, username, ceratime, creatuser），forum\_info表（info\_id, info\_reply, info\_see, info\_lastuser, info\_lastime, main\_id）,forum\_main表（main\_id, main\_title, main\_flag, main\_type, main\_content, main\_creatime, main\_creatuser, main\_recommend, main\_delete, main\_nickname, main\_zan）forum\_second表（sec\_id, main\_id, sec\_sequence, sec\_content, sec\_creatuser, sec\_creatime, sec\_nickname, sec\_resequence, sec\_delete, sec\_reid, username）,my\_main表（main\_id，main\_title，main\_content，main\_creatime，main\_creatuser，main\_commend ），mysecond表（sec\_id，main\_id，sec\_content，sec\_creatime，sec\_creatuser，sec\_sequence），sys\_login（login\_id，username，password，email，wxname），sys\_role表（role\_id，role\_name，role\_des，role\_pid）。

### 1.5.2 查看与回复帖子模块相关类实现

1.主贴信息类forumMain：把有关主贴的信息与操作封装在forumMain类中，forumMain类的成员变量和方法如下：

**public** **class** forumMain {

**private** String mainId;

**private** String mainTitle;

**private** String mainFlag;

**private** String mainType;

**private** Date mainCreatime;

**private** String mainCreatuser;

**private** Integer mainRecommend;

**private** String mainDelete;

**private** String mainNickname;

**private** Integer mainZan;

**private** String mainContent;

**public** String getMainId()

**public** **void** setMainId(String mainId)

**public** String getMainTitle()

**public** **void** setMainTitle(String mainTitle)

**public** String getMainFlag()

**public** **void** setMainFlag(String mainFlag)

**public** String getMainType()

**public** **void** setMainType(String mainType)

**public** Date getMainCreatime()

**public** **void** setMainCreatime(Date mainCreatime)

**public** String getMainCreatuser()

**public** **void** setMainCreatuser(String mainCreatuser)

**public** Integer getMainRecommend()

**public** **void** setMainRecommend(Integer mainRecommend)

**public** String getMainDelete()

**public** **void** setMainDelete(String mainDelete)

**public** String getMainNickname()

**public** **void** setMainNickname(String mainNickname)

**public** Integer getMainZan()

**public** **void** setMainZan(Integer mainZan)

**public** String getMainContent()

**public** **void** setMainContent(String mainContent)

}

2.回帖信息类forumSecond：把对主贴的回帖信息与操作封装在forumSecond类中，与数据库中forum\_second表对应。此类的成员变量与方法如下：

**public** **class** forumSecond {

**private** String secId;

**private** String mainId;

**private** Integer secSequence;

**private** String secCreatuser;

**private** Date secCreatime;

**private** String secNickname;

**private** Integer secResequence;

**private** String secDelete;

**private** String secContent;

**public** String getSecId()

**public** **void** setSecId(String secId)

**public** String getMainId()

**public** **void** setMainId(String mainId)

**public** Integer getSecSequence()

**public** **void** setSecSequence(Integer secSequence)

**public** String getSecCreatuser()

**public** **void** setSecCreatuser(String secCreatuser)

**public** Date getSecCreatime()

**public** **void** setSecCreatime(Date secCreatime)

**public** String getSecNickname()

**public** **void** setSecNickname(String secNickname)

**public** Integer getSecResequence()

**public** **void** setSecResequence(Integer secResequence)

**public** String getSecDelete()

**public** **void** setSecDelete(String secDelete)

**public** String getSecContent()

**public** **void** setSecContent(String secContent)

}

3.查看与回复帖子控制器类SecondController，实现了子贴页面的初始化与回帖功能。此类中方法ModelAndView viewContro是初始化子贴页面，将从数据库获取的当前子贴页面中的帖子内容、回帖内容、帖子的分页导航等部件返回给jsp文件；方法ModelAndView saveSecond是保存用户发帖内容到数据库，同时刷新当前 ，将用户的回帖信息更新至当前页面。此类的成员变量与方法如下：  
@Controller

**public** **class** SecondController {

@Resource

ForumServiceImpl fS;

@Resource

SimpleDateFormat simpleDateFormat;

@RequestMapping(value="/secondContent")

**public** ModelAndView viewContro (HttpServletRequest request,String mainId,@RequestParam(name="order",defaultValue="ASC") String order,@RequestParam(name="page",defaultValue="1") Integer page,@RequestParam(name="row",defaultValue="15")Integer row)

@RequestMapping(value="/saveSencondContent")

**public** ModelAndView saveSecond (HttpServletRequest request,String content,String mainId,String reId,@RequestParam(name="resequence",defaultValue="0")Integer resequence)

}

4.查看与回复帖子内容服务接口类ForumServiceImpl，此类包含子贴中与服务器相关数据的增删改查操作。方法getSecondMain获取详细的帖子信息；方法getSecond获取回帖信息；方法getSecondCount获取回帖总数；方法getPage子贴页面的分页导航信息。此类的成员变量与部分相关方法如下（因为此类中还包含其他模块所需的与数据库操作的方法，在此略去）：

@Service

**public** **class** ForumServiceImpl{

@Resource

forumMainMapper fmm;

@Resource

JdbcTemplate jdbcTemplate;

@Resource

SimpleDateFormat simpleDateFormat;

/\*\*

\* 获取详细的帖子主贴

\* **@param** mainId

\* **@return**

\*/

**public** Map<String,Object> getSecondMain (String mainId)

/\*\*

\* 获取跟帖

\* **@param** mainId

\* **@param** order

\* **@param** start

\* **@param** offset

\* **@return**

\*/

**public** List<Map<String,Object>> getSecond (String mainId,String order,Integer start,Integer offset)

**public** **long** getSecondCount (String mainId)

/\*\*

\* 如果传递过来的mainType 证明这是一个2级菜单

\* 如果传递过来的是mainId 证明这是一个3级菜单

\* **@param** mainType

\* **@param** mainId

\* **@return**

\*/

**public** Map<String,Object> getPath2 (String mainType,String mainId)

}

/\*\*

\* 获取模块分类

\* **@return**

\*/

**public** List<Map<String,Object>> getDictionaryByGroup (String group)

**public** String getPage (Long count,Integer currentPage,Integer offset,Map<String,String> parm)

}

4.查看与回复帖子的动态网页文件secondContent.jsp:实现网站主页动态初始化。

5.各个跳转的子页面，如mainContent.jsp、secondContent等。

### 1.5.3 查看与回复帖子模块文件及跳转关系

（1）模块涉及代码文件列表如表1.11所示。

表1.11 查看与回复帖子模块涉及代码文件列表

|  |  |  |
| --- | --- | --- |
| **文件名** | **文件路径** | **文件说明** |
| index.jsp | F:\javaproject\mrbbs\WebContent\WEB-INF\view | 网站动态主页面 |
| mainContent.jsp | F:\javaproject\mrbbs\WebContent\WEB-INF\view | 网站动态子版块页面 |
| secondContent.jsp | F:\javaproject\mrbbs\WebContent\WEB-INF\view | 网站动态查看与回复帖子页面 |
| Jinghua.jps | F:\javaproject\mrbbs\WebContent\WEB-INF\view | 网站动态精华帖集中查看页面 |
| Login.jsp | F:\javaproject\mrbbs\WebContent\WEB-INF | 网站注册与登录页面 |

（2）模块涉及页面跳转关系图，如图1.18所示。



图1.18主页面模块涉及界面跳转关系图

### 1.5.4查看与回复帖子模块程序流程

查看与回复帖子模块实现程序流程如图1.19所示。



图1.19 查看与回复帖子模块程序流程图

### 1.3.5 查看与回复帖子模块实现界面

查看与回复帖子模块实现界面如图1.20所示。

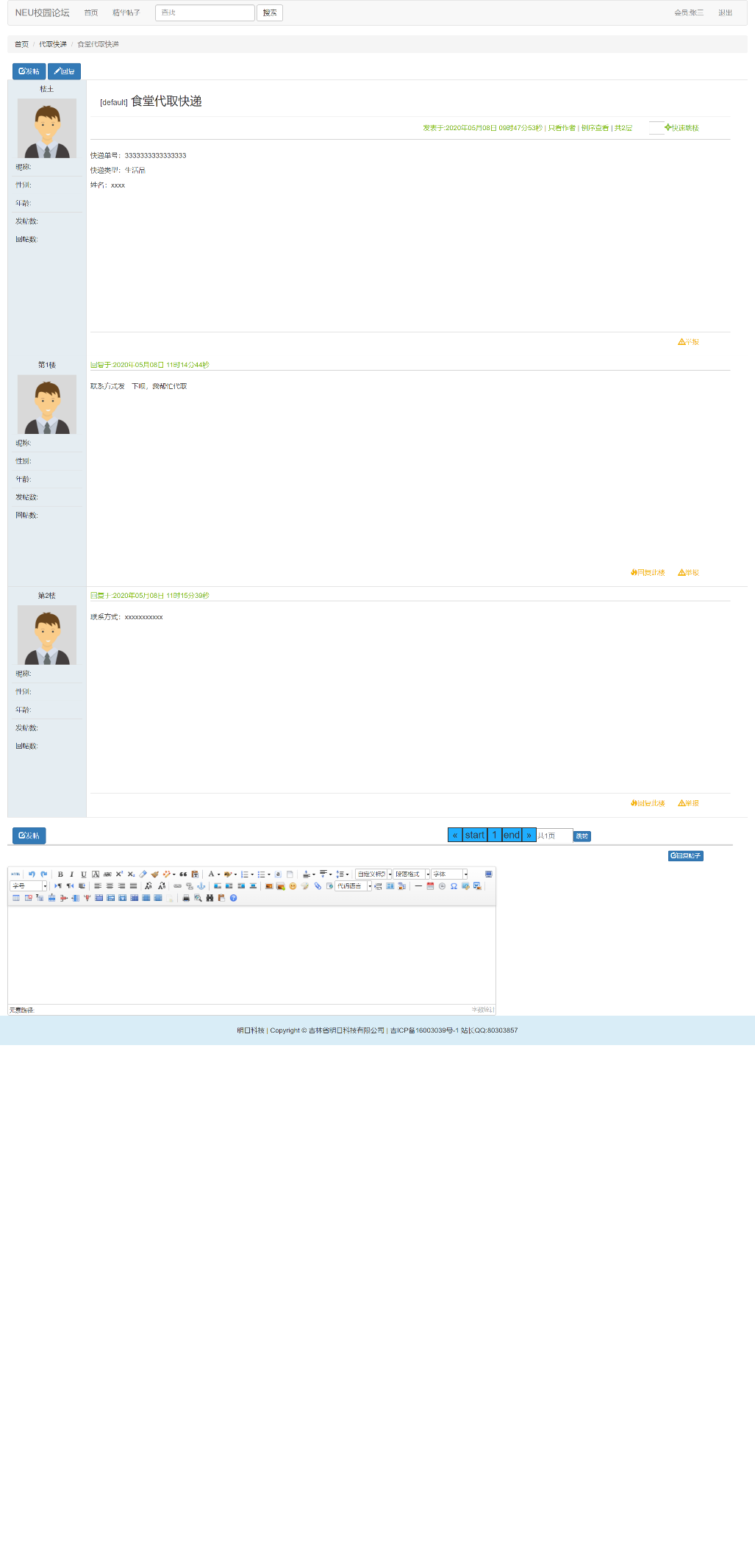


图1.20查看与回复帖子功能区实现界面

## 1.6 用户登录和退出模块实现

### 1.6.1 用户登录和退出模块实现简介

本团队的用户登录和退出模块基于Apache Shiro安全框架，使用该框架提供的身份识别、访问控制、密码加密的功能，对本网站的用户登录进行控制。本网站用户的退出方式有两种：①用户手动退出②用户关闭浏览器退出。其中在实现用户登录功能时会对用户输入的用户名、密码、验证码与数据库中内容进行比对，并设置当失败次数过多时给出提示，锁定账号。用户成功登录后会跳转到网站首页，可以在首页进行进一步的操作。本模块涉及到的数据库主要有：base\_userinfo、sys\_login、sys\_login\_role。

### 1.6.2 用户登录和退出模块相关类实现

用户登录所涉及到的类为UserLogin。使用到该类中的方法有：

方法一：

public Map<String,String> login (HttpServletRequest request,String username,String password,String verifyCode)

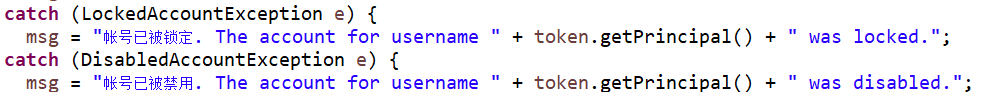
经过判断用户的输入，如果用户输入的密码错误给出提示：



如果用户的输入失败次数过多给出提示：



用户账号被锁定或者被禁用给出提示：



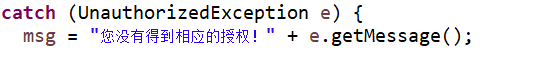
用户账号登录时间过久过期给出提示：



用户账号不存在给出提示：



用户没有被授权给出提示：



方法二：

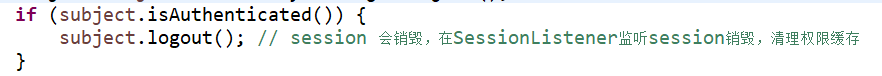
public ModelAndView login(HttpServletRequest request,String username,String password,@RequestParam(defaultValue="0000") String verifyCode, Model model)

该方法用于判断用户输入的登录信息是否正确。

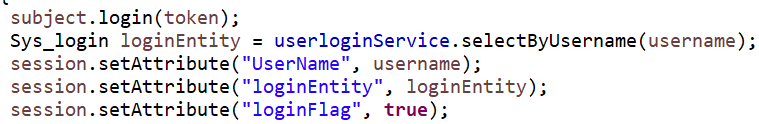
对用户输入的验证码进行判断：



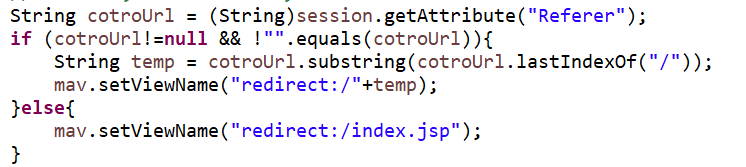
判断用户是否已经登录：



用户成功登录后，对相关信息进行保存，将用户登录状态置为true：



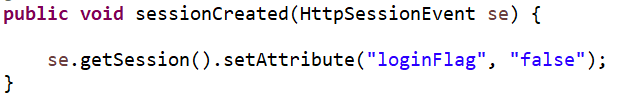
成功登录进行页面跳转：



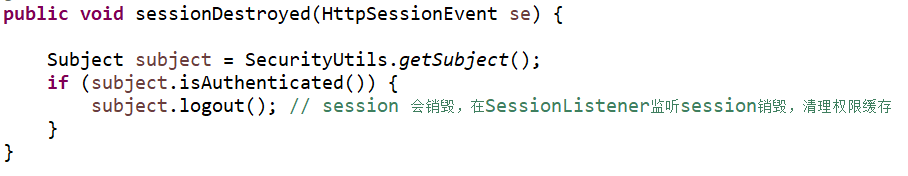
登录失败时对用户失败原因进行输出提示，并返回登录页面：



用户退出所用到的类为SessionListenerBySeachLogin。在该类中把用户的登录状态由true置为false，表示用户登录状态由登录转变为退出：



并对session的值进行销毁：



### 1.6.3 用户登录和退出模块文件及跳转关系

（1）模块涉及代码文件列表如表1.12所示。

表1.12 用户注册模块涉及代码文件列表

|  |  |  |
| --- | --- | --- |
| **文件名** | **文件路径** | **文件说明** |
| login.jsp | mrbbs\WebContent\WEB-INF\login.jsp | 实现登录界面，包含对于登录操作的JavaScript函数定义 |
| SessionListenerBySeachLogin.java | mrbbs\Java Resources\src\  com.mrkj.ygl.listener\SessionListenerBySeachLogin.java | 实现用户账号退出 |
| UserLogin.java | mrbbs\Java Resources\src\  com.mrkj.ygl.web\UserLogin.java | 实现用户登录的条件判断，与错误提示信息输出 |
| Index.jsp | mrbbs\WebContent\WEB-INF\view\myJSP\index.jsp | 网站首页布局设计与显示 |

（2）模块涉及页面跳转关系图，如图1.21所示。

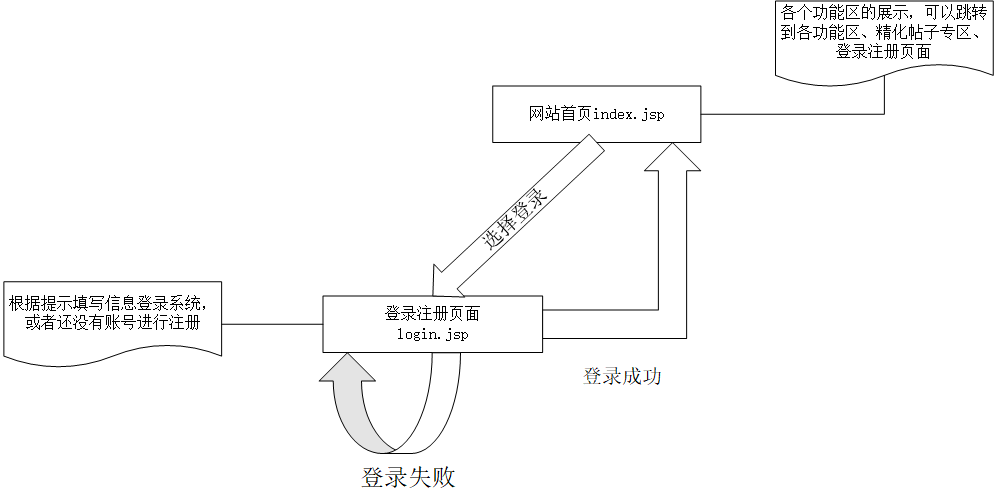


图1.21 用户登录和退出涉及界面跳转关系图

### 1.6.4 用户登录和退出模块程序流程

用户登录模块实现程序流程如图2.2所示。用户进入登录界面进行登录操作，由于本网站没有使用验证码，所以验证码的验证是系统默认验证，当用户输入账户和密码后系统对比数据库中的内容进行判断，并对该用户的登录状态进行判断，如果已经登录则销毁当前变量值，如果未登录判断用户账号和密码输入正确则登录成功跳转到网站首页，如果登录失败，这其中的原因可能是多样的，分别给出合适的提示，并返回登录界面再次输入账户和密码进行登录。

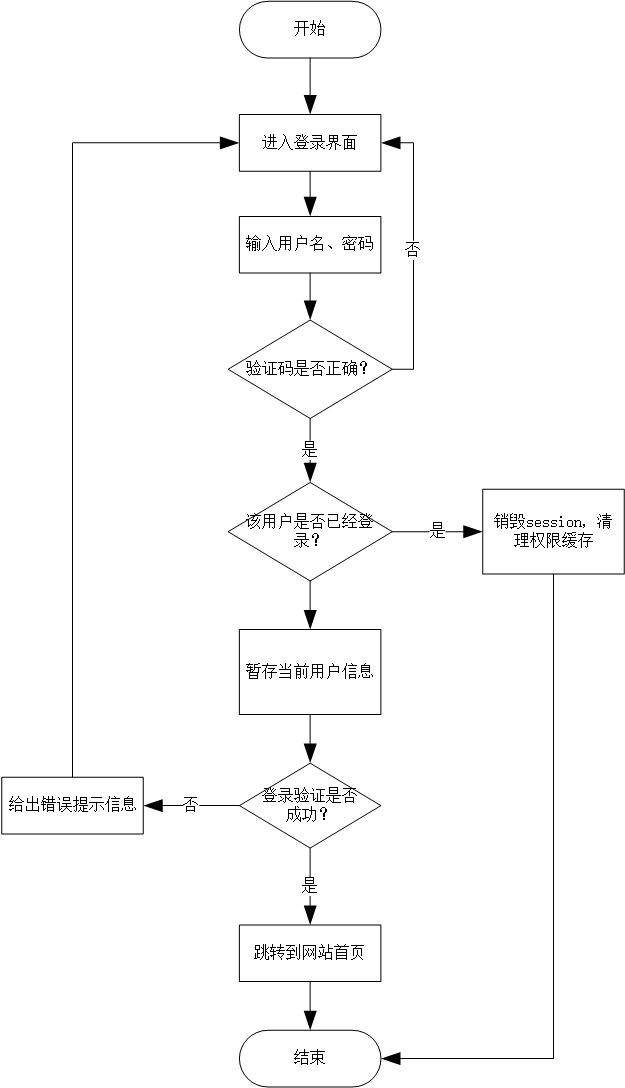


图1.22 用户登录模块程序流程图

### 1.6.5 用户登录和退出模块实现界面

用户注册与登录界面如图所示：

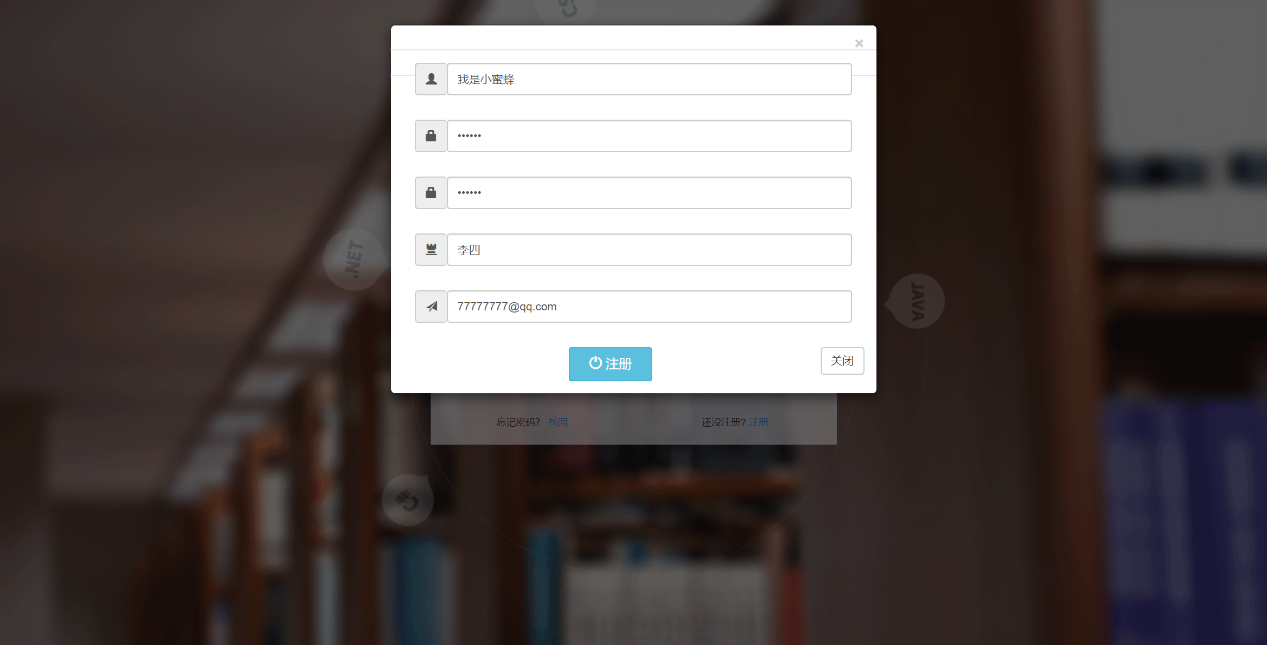


图1.23用户注册实现界面



图1.24用户登录实现界面

## 1.7 实现过程中的需求和设计变动情况说明

1.用户角色与权限

在需求分析阶段和系统设计阶段，此网站的用户有游客（未登录者）、学生（注册登录且经过学邮箱认证）、商户（注册登录且经过身份认证）、学生组织负责人（注册登录且经过身份认证）、网站管理员（注册登录且经过身份认证）。每个角色拥有不同的权限。游客仅能浏览INFORMATION信息区中的学校资料等帖子，学生可以使用除网站管理员外所有的功能，商户能在LIFE生活区浏览和发布帖子，学生组织负责人除了拥有普通学生的权限外，还能在INFORMATION信息区发布有关活动竞赛的帖子。

在迭代的开发过程中，网站角色做了部分简化。到此开发时间截止时，网站的角色实现了两种：登录者和未登录者，他们的权限如下图所示：



图1.25 系统角色权限图

2.数据库设计

由于网站的角色与权限做出了一定的简化，因此在系统设计部分所设计的数据库，也相应做出了一定的调整。原先的数据库中，游客、学生、商户、学生组织负责人和网站管理员都各自拥有一张关系表，来存储相应账号的信息。

在最后发布的可用版本中，有关网站用户的的关系表合并了有关学生、商户、学生组织负责人和网站管理员的数据，即将他们统一看做网站的登录用户进行账户信息管理。

3.网站的管理与维护界面

由于迭代开发时间紧迫，因此首先实现了网站最核心的功能。原先设计的扩展功能——网站的管理与用户页面未实现，如果需要维护网站，需要对代码进行调试和维护，如果未来需要对此网站进行改进，此方面是可以改进的方向。

## 1.8 实现小结

软件开发中，其最终目的是得到符合用户需求与开发商利益的程序源代码以及可执行系统。而软件开发的实施活动的目标就是将系统“做出来”，它包括编码、单元测试和集成。是软件开发过程中必不可少且具有重要意义的环节。

在本次的“LSIC社区”网站的实施活动中，我们依然采用了RUP过程模型、敏捷方法XP和敏捷方法Scrum相结合的模型开发方法。首先确定网站的基本功能：注册、登录与浏览、发布帖子，并将网站开发分为前端与后端，依据此为团队中各个角色分配任务。并采取了敏捷方法XP的开发方法，即结对开发，两两一组，一个人负责编写代码，一个人负责进行测试。例如，在本次开发过程中，由黄菊、侯宝玉、刘畅进行后端开发，由徐建伟、赵肖进行前端开发，并将前端与后端开发人员两两结对（人员有重复），以达到提高开发效率、前后端开发相互了解业务流程的目的。

经过学习网站开发相关知识学习、租用服务器、购买域名、搭建开发环境、任务分配、编写代码、单元测试与集成后，本团队最终完成了“LSIC”社区网站的开发，但由于服务器租用时间过短，不能解析域名，最终的项目只能通过IP地址进行访问，且5.10服务器租用时间到期后，只能本地运行此项目，这是此项目的遗憾之一。除此之外，若未来能够进一步对此项目进行改进，网站的角色与权限可以进一步丰富，这也是本次开发的不足之一。

# 2 系统测试

## 2.1人员安排

“LSIC”社区网站系统测试小组

组长：徐建伟

组员：赵肖

黄菊

侯宝玉

刘畅

## 2.2 系统通用类测试

### 2.2.1 数据库通用类测试

（1）白盒测试用例

数据库通用白盒测试用例如表2.1所示。

表2.1 数据库通用白盒测试用例表

|  |  |
| --- | --- |
| **用例名称** | **发布新消息测试用例** |
| 用例id | Test-0011 |
| 基本描述 | 用户将数据上传至服务器数据库。 |
| 测试方案 | 遍历所有用户上传数据至服务器数据库端的函数中的分支。 |
| 输入数据 | 1. 系统接收用户的用户名和密码。 2. 系统上传帖子至服务器。 3. 系统上传各类资料至服务器。 4. 系统接收帖子。 5. 系统接收各类资料 |
| 预期结果 | 第一组测试正确执行，用户名和密码成功上传至服务器。  第二组测试系统正确执行，可以在本地帖子栏看到。  第三组测测试系统正确执行，可以在本地资源栏看到。  第四组测试系统正确执行，可以在另一台设备中查看发布过的帖子。  第五组测试系统正确执行，可以在另一台设备上查看上传过的资料。 |

（2）黑盒测试用例

数据库通用黑盒测试用例如表2.2示。

表2.2数据库通用黑盒测试用例表

|  |  |
| --- | --- |
| **用例名称** | **数据库上传服务器测试用例** |
| 用例id | Test-0012 |
| 基本描述 | 用户将数据上传至服务器数据库。 |
| 测试方案 | 分别测试学生用户，管理员用户，组织负责人用户在涉及到上传数据至服务器数据库时可能遇到的问题。 |
| 输入数据 | 1. 学生用户输入特殊字符串当做用户名、密码。 2. 用户上传空帖子到服务器 3. 用户上传空资料到服务器 |
| 预期结果 | 第一组测试正确执行，，用户名和密码成功上传至服务器，用户可以正常登陆。  第二组测试系统提示帖子为空，请输入帖子。  第三组测试系统提示资源为空，请选择资源。 |

### 2.2.2 网络连接通用类测试

（1）白盒测试用例

网络连接通用白盒测试用例如表2.3所示

表2.3网络连接通用白盒测试用例表

|  |  |
| --- | --- |
| **用例名称** | **网络连接通用白盒测试用例** |
| 用例id | Test-0013 |
| 基本描述 | 系统在运行期间可能会出现的各种问题。 |
| 测试方案 | 对系统提供的各项功能进行测试。 |
| 输入数据 | 1. 登录后选择发布帖子。 2. 非登录直接进入生活模块发布帖子 3. 非登录直接进入学习模块发布帖子 4. 非登录直接进入资源模块发布帖子 5. 非登录直接进入交流模块发布帖子 |
| 预期结果 | 第一组测试正常进行。  第二组测试系统提示无权限，可正常查看帖子，但不能发布帖子。  第三组测试系统提示无权限，可正常查看帖子，但不能发布帖子。  第四组测试系统提示无权限，可正常查看帖子，但不能发布帖子。  第五组测试系统提示无权限，可正常查看帖子，但不能发布帖子。 |

（2）黑盒测试用例

网络连接通用黑盒测试用例如表2.4所示

表2.4网络连接通用黑盒测试用例表

|  |  |
| --- | --- |
| **用例名称** | **网络连接通用白盒测试用例** |
| 用例id | Test-0014 |
| 基本描述 | 用户在使用软件时可能发生的一系列问题。 |
| 测试方案 | 模拟用户使用，在使用中发现可能出现的问题。 |
| 输入数据 | 1. 下载资源时断开网络链接。 2. 播放视频或进行测试时关闭网页 3. 下载资源时断开网络链接。 4. 在网络断开时浏览帖子 |
| 预期结果 | 第一组测试提示用户未连接网络，同时返回至主界面。  第二组测试系统正常结束，无错误。  第三组测试系统无法正常下载，需重新打开软件重新下载。  第四组测试系统正确执行，但会提示网络已断开，无法加载。 |

### 2.2.3 系统通用类测试报告

系统通用类测试报告如表2.5所示。

表2.5 系统通用类测试报告

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 功能 | 错误描述 | | | 统计项 | | | | |
| 应达目标 | 实际结果 | 出现错误条件（几率） | 错误类型 | 测试者 | 测试日期 | 改正日期 | 状态 |
| 1 | 学生上传空帖子 | 系统提示为空 | 正常上传 | 100% | 功能错误 | 徐建伟 | 2020.5.7 | 2020.5.7 | 完成 |
| 2 | 下载资源 | 在重新连接网络后应继续下载 | 只能重新下载 | 100% | 功能错误 | 徐建伟 | 2020.5.7 | 202.5.8. | 完成 |
| 3 | 进入各个模块 | 在未登录情况下直接进行测试 | 未登录可以直接进入模块 | 0% | 无 | 徐建伟 | 2020.5.7 | 无 | 无 |

## 2.3生活模块测试

用户通过生活模块进行代取快递，二手交易，组团拼单，拼车，出游兼职等，是本软件的主要功能模块。

其中主要包括四个子模块，每个子模块还有各自的功能。

代取快递子模块：包含管理代取请求，删除代取请求，领取代取请求等功能。

主要功能是发布与代取快递有关的信息，学生可以考虑是否领取快递，发布者还可以删除代取快递信息，提供学生间交流的平台，询问相关信息。

二手交易子模块：包含管理二手货物请求，删除二手货物信息，领取二手货物请求等功能。主要功能是发布与二手交易有关的信息，学生可以考虑是否领取进行交易，发布者还可以删除二手货物信息，提供学生间交流的平台，询问相关信息。

组团拼单/拼车/出游子模块：包含管理组团拼车/拼单/出游信息，删除组团拼车/拼单/出游信息，查看组团拼车/拼单/出游信息等功能。主要功能是发布组团拼车/拼单/出游信息，学生可以考虑是否组团，发布者还可以删除组团拼车/拼单/出游信息，提供学生之间交流的平台，查看及询问相关信息。

兼职子模块：包含管理管理兼职信息，删除兼职信息，查看兼职信息。主要功能是发布兼职信息信息，学生可以考虑是否兼职，发布者还可以删除兼职信息，提供学生之间交流的平台，查看及询问相关信息。

### 2.3.1 生活模块白盒测试用例

生活模块白盒测试用例如表2.6所示。

表2.6生活模块白盒测试用例表

|  |  |
| --- | --- |
| **用例名称** | **生活模块白盒测试用例** |
| 用例id | Test-0021 |
| 基本描述 | 用户发布帖子并查看帖子 |
| 测试方案 | 模拟用户发帖、查看帖子以及交流的过程。 |
| 输入数据 | 1. 在网络正常时发布帖子。 2. 在网络较差或断开时发布帖子。 3. 在网络正常浏览帖子 4. 在网络异常时浏览帖子 |
| 预期结果 | 第一组测试可以正常发布帖子。  第二组测试提示网络较差或未链接，无法发布帖子。  第三组测试可以正常浏览帖子  第四组测试系统提示网络异常，无法浏览帖子 |

### 2.3.2 生活模块黑盒测试用例

表2.7生活模块黑盒测试用例表

|  |  |
| --- | --- |
| **用例名称** | **网络连接通用白盒测试用例** |
| 用例id | Test-0022 |
| 基本描述 | 模拟未登录用户发帖删帖等 |
| 测试方案 | 模拟未登录用户使用，在使用中发现可能出现的问题。 |
| 输入数据 | 1. 用户未登录进行发帖 2. 用户未登录查看帖子 3. 用户登录后查看帖子 |
| 预期结果 | 第一组测试提示用户未登录，不能发帖。  第二组测试用户可以查看帖子。  第三组测试用户可以正常查看帖子 |

### 2.3.3 生活模块测试报告

生活模块测试报告如表2.8所示。

表2.9 生活模块测试报告

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 功能 | 错误描述 | | | 统计项 | | | | |
| 应达目标 | 实际结果 | 出现错误条件（几率） | 错误类型 | 测试者 | 测试日期 | 改正日期 | 状态 |
| 1 | 网络较差情况下发布帖子 | 不能进行发布 | 有时会出现系统崩溃 | 10% | 网络连接错误 | 徐建伟 | 2020.5.6 | 2020.5.8 | 完成 |
| 2 | 用户未登录时发布帖子 | 提示用户未登录，请登陆后进行发布 | 出现用户可以发布帖子 | 100% | 功能错误 | 徐建伟 | 2020.5.6 | 2020.5.8 | 完成 |
| 3 | 用户未登录时查看帖子 | 用户可以正常查看帖子 | 用户可以正常查看 | 0% | 无 | 徐建伟 | 2020.5.6 | 无 | 完成 |

### 2.3.4 生活模块调试过程

1. 在网络较差的情况下发布帖子时，有时会出现发布失败的情况，由于之前未设置异常处理所以会导致程序直接崩溃，在加入异常处理后提示用户网络断开连接，不能进行发布。

2.因为用户在未登录时也可以发布帖子，我们发现在编程时没有对发帖进行限制，于是后来在改正时加入了首先要进行登录，之后才可以进行发帖的功能

## 2.4 学习模块测试

用户通过习题模块进行编辑帖子，上传相关资料，删除帖子，浏览帖子，回复别人，下载资料等。

其中主要包括三个子模块，每个子模块还有各自的功能。

发帖（考研/保研相关）子模块：经过身份验证的学校学生，在此模块，编辑帖子的内容并发布，内容包括文字、表情等，学生在发布的帖子中，上传资料，资料类型应当包括文档、图片、视频等 。

删帖子模块：学生在任何时候，都可以删除自己发布过的帖子。

回帖子子模块：学校学生浏览所有被发布的帖子。帖子按照时间排序，此外，也可以按照回帖数等方式排序的功能；学校学生可以在某个帖子中回复别人；学校学生可以下载帖子中的资料到本地。

### 2.4.1学习模块白盒测试用例

学习模块白盒测试用例如表2.10所示。

表2.10学习模块白盒测试用例表

|  |  |
| --- | --- |
| **用例名称** | **学习模块白盒测试用例** |
| 用例id | Test-0041 |
| 基本描述 | 用户上传各类文件 |
| 测试方案 | 模拟用户上传各类文件 |
| 输入数据 | 1. 在网络正常时上传文件 2. 在网络较差时上传文件 3. 在网络断开时上传文件 |
| 预期结果 | 第一组测试可以正常上传  第二组测试上传时速度缓慢，在网络非常不好时，会出现上传失败，再次链接网络时要重新上传  第三组测试系统提示网络未链接，无法上传 |

### 2.4.2 学习模块黑盒测试用例

表2.11学习模块黑盒测试用例表

|  |  |
| --- | --- |
| **用例名称** | **学习模块白盒测试用例** |
| 用例id | Test-0042 |
| 基本描述 | 用户上传各类压缩包文件 |
| 测试方案 | 模拟用户上传各类压缩包文件，看是否可以上传成功 |
| 输入数据 | 1. 用户上传.zip文件 2. 用户上传.rar文件 |
| 预期结果 | 第一组测试系统提示用户不能上传  第二组测试系统提示用户不能上传 |

### 2.4.3 学习模块测试报告

学习模块测试报告如表2.12所示。

表2.12 学习模块测试报告

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 功能 | 错误描述 | | | 统计项 | | | | |
| 应达目标 | 实际结果 | 出现错误条件（几率） | 错误类型 | 测试者 | 测试日期 | 改正日期 | 状态 |
| 1 | 网络较差情况下上传文件 | 上传速度缓慢，网络断开后停止，网络重新连接后继续上传 | 网络断开后显示上传失败，再次链接后不可继续上传 | 100% | 网络连接错误 | 徐建伟 | 2020.5.5 | 2020.5.7 | 完成 |
| 2 | 网络断开时上传文件 | 提示用户网络断开，不能上传 | 系统提示网络断开，不能上传 | 0% | 无 | 徐建伟 | 2020.5.5 | 2020.5.8 | 完成 |
| 3 | 用户上传.zip文件 | 系统提示用户不能上传 | 系统崩溃 | 5% | 功能错误 | 徐建伟 | 2020.5.5 | 2020.5.7 | 完成 |

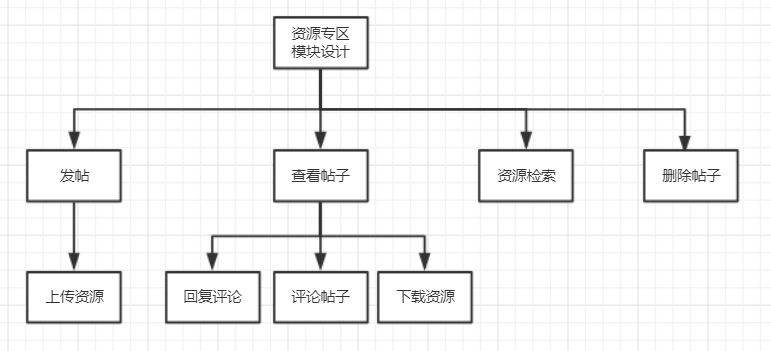
### 2.4.4 学习模块调试过程

1.一开始网络断开时即出现上传失败，因为我们系统没有对其进行缓存，后来加入了上传时即进行缓存的功能，下次网络连接好时即可接着之前的继续上传

2. 用户上传.zip文件时有时会出现系统崩溃，也是由于之前未设置异常处理所以会导致程序直接崩溃，后来设置异常处理后系统即可提示用户不能上传

## 2.5 资源模块测试

用户通过资源专区模块利用发帖的形式进行资源的上传和下载，并且支持对于资源的检索。详细模块图如下：

图2.1资源专区模块设计详细图

其中主要包括四个子模块，每个子模块的详细介绍如下：

发帖子模块：用户在登录后可以使用发帖功能，帖子包括标题、标签、内容、资源。在发帖的过程中，标题和资源是必填项。

查看帖子子模块：用户可以查看帖子的内容，并且进行评论和回复评论。同时用户可以对帖子中的资源进行下载。

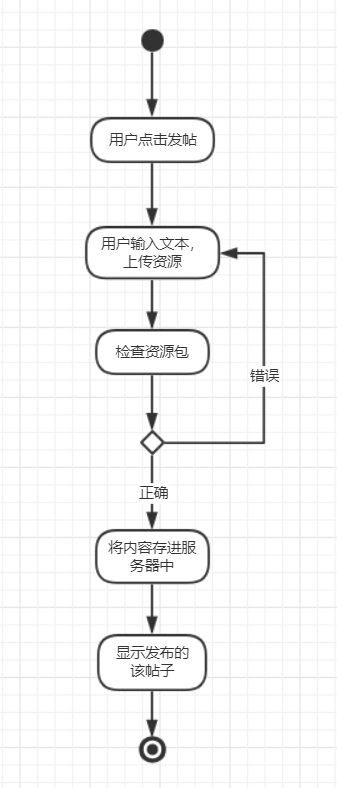
资源检索子模块：可以对现有的资源进行检索。

删除帖子子模块：对于自己发布的帖子，用户可以选择删除。删除后，帖子中的资源会一并删除。

### 2.5.1 资源模块白盒测试用例

由于待测试用例较多，这里仅针对该模块的发帖功能写出一组测试用例。

资源专区模块发帖功能流程图如下：

图2.2资源专区模块发帖功能流程图

资源专区模块白盒测试用例如表2.13所示

表2.13资源专区模块白盒测试用例

|  |  |
| --- | --- |
| **用例名称** | **资源专区模块白盒测试用例** |
| 用例id | Test-0021 |
| 基本描述 | 网站保存用户输入的帖子内容和资源。 |
| 测试方案 | 测试在用户在发帖过程中可能出现的所有情况。 |
| 输入数据 | 1. 发帖过程正常。 2. 发帖过程中网速波动。 3. 发帖过程中资源路径修改。 |
| 预期结果 | 第一组测试正确执行，数据库成功更新。  第二组测试执行失败，显示资源上传超时，上传失败。  第三组测试执行成功，但是保存的资源无效。 |

### 2.5.2 资源模块黑盒测试用例

由于待测试用例较多，这里仅针对该模块的发帖功能写出一组测试用例。

资源专区模块黑盒测试用例如表2.14所示

表2.14资源专区模块黑盒测试用例

|  |  |
| --- | --- |
| **用例名称** | **资源专区模块黑盒测试用例** |
| 用例id | Test-0022 |
| 基本描述 | 用户点击发帖，运行发贴子模块。 |
| 测试方案 | 测试在用户在发帖过程中可能产生的行为以及发生的结果。 |
| 输入数据 | 1. 用户在发帖过程中关闭网站。 2. 用户在资源上传过程中暂停或取消。 3. 用户未上传资源。 |
| 预期结果 | 第一组测试执行失败，发帖的内容丢失。  第二组测试正确执行，显示资源上传失败。  第三组测试正确执行，帖子无资源存在。 |

### 2.5.3 资源模块测试报告

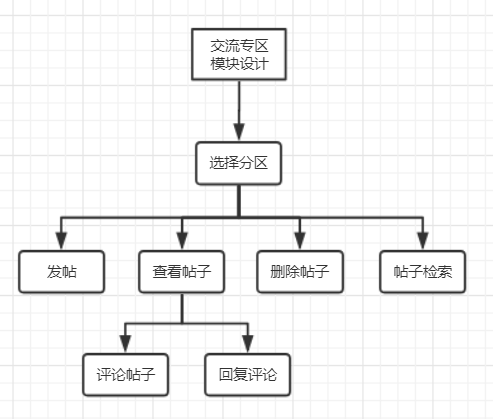
资源专区模块测试报告如表2.15所示

表2.15 资源专区模块测试报告

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 功能 | 错误描述 | | | 统计项 | | | | |
| 应达目标 | 实际结果 | 出现错误条件（几率） | 错误类型 | 测试者 | 测试日期 | 改正日期 | 状态 |
| 1 | 发帖过程中退出网站 | 已输入部分上传至服务器 | 所有结果都未上传 | 100% | 功能错误 | 赵肖 | 2020.5.2 | 无 | 待改 |
| 2 | 未上传资源或者上传失败 | 系统提示重新上传资源 | 发帖成功但无有效资源 | 100% | 功能错误 | 赵肖 | 2020.5.2 | 2020.5.4 | 完成 |

## 2.6 交流模块测试

用户通过交流专区模块利用发帖的形式与其他人进行交流，并且设有东大资料、东大报纸、活动\竞赛三个分区。详细模块图如下：

图2.3交流专区模块设计详细图

模块包含了三个主题分区，不同类型的帖子在相对应的分区中发布。在不同的分区中，功能相同，区别只在于帖子的内容主题。模块的详细子模块介绍如下：

发帖子模块：用户在登录后可以使用发帖功能，帖子包括标题、标签、内容。在发帖的过程中，标题是必填项。这个子模块和资源专区模块的区别是没有资源相关的功能，在发帖过程中的限制也更少一些。

查看帖子子模块：用户可以查看帖子的内容，并且进行评论和回复评论。此模块的功能和资源专区模块的大致相同。

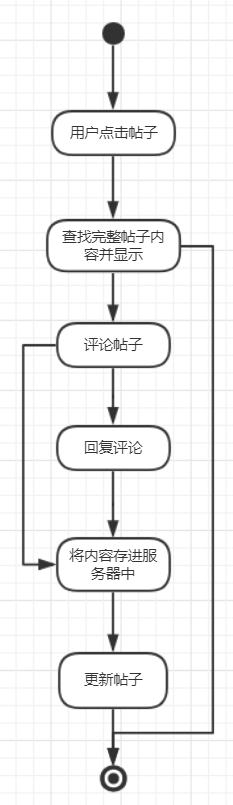
删除帖子子模块：对于自己发布的帖子，用户可以选择删除。这个子模块和资源专区模块的区别是没有附带的资源，更加简单。

帖子检索子模块：用户可以对分区中的帖子进行检索，但是不能对在此分区中对其他分区进行检索。

### 2.6.1 交流模块白盒测试用例

由于待测试用例较多，这里仅针对该模块的查看帖子功能写出一组测试用例。

交流专区模块查看帖子功能流程图如下：

图2.4交流专区模块查看帖子功能流程图

交流专区模块白盒测试用例如表2.16所示

表2.16交流专区模块白盒测试用例

|  |  |
| --- | --- |
| **用例名称** | **交流专区模块白盒测试用例** |
| 用例id | Test-0031 |
| 基本描述 | 网站查找帖子内容并显示。 |
| 测试方案 | 测试在用户在查看帖子过程中可能出现的所有情况。 |
| 输入数据 | 1. 用户在帖子更新过程中的网络波动 2. 用户输入的内容中包含敏感词或者非法内容。 3. 用户在评论未完成时关闭网站。 |
| 预期结果 | 第一组测试正确执行，数据库能成功更新，但也有可能显示不全。  第二组测试正确执行，敏感词用\*代替。  第三组测试执行失败，未保存的内容丢失。 |

### 2.6.2 交流模块黑盒测试用例

由于待测试用例较多，这里仅针对该模块的查看帖子功能写出一组测试用例。

交流专区模块黑盒测试用例如表2.17所示

表2.17交流专区模块黑盒测试用例

|  |  |
| --- | --- |
| **用例名称** | **交流专区模块黑盒测试用例** |
| 用例id | Test-0032 |
| 基本描述 | 用户点击帖子，转到帖子界面显示完整帖子。 |
| 测试方案 | 测试在用户在查看帖子过程中可能产生的行为以及发生的结果。 |
| 输入数据 | 1. 用户连续多次点击评论按钮。 2. 用户未评论完成点击其他页面。 |
| 预期结果 | 第一组测试执行成功，显示第一次点击的操作。  第二组测试出现错误，评论的内容丢失。 |

### 2.6.3 交流模块测试报告

交流专区模块测试报告如表2.18所示

表2.18 资源专区模块测试报告

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 功能 | 错误描述 | | | 统计项 | | | | |
| 应达目标 | 实际结果 | 出现错误条件（几率） | 错误类型 | 测试者 | 测试日期 | 改正日期 | 状态 |
| 1 | 评论过程中保存内容 | 评论未完成跳转到其他页面仍然保存 | 评论内容消失 | 100% | 功能错误 | 赵肖 | 2020.5.4 | 无 | 待改 |
| 2 | 点击按钮即时反应 | 连续多次点击实现第一次的操作 | 网站卡顿 | 100% | 功能错误 | 赵肖 | 2020.5.3 | 2020.5.4 | 完成 |
| 3 | 跳转到帖子界面 | 正确显示 | 网络波动中，界面显示不全 | 9.2% | 功能错误 | 赵肖 | 2020.5.3 | 2020.5.5 | 完成 |

## 2.7 系统集成测试

集成测试是在每个模块完成单元测试以后，将所有功能模块集成在一起的测试，以验证各模块的正确性和接口的正确性的测试。在单元测试的基础上，集成测试将所有的软件单元按照概要设计规格说明的要求组装成模块，并检测各部分工作是否达到或实现相应技术指标及要求，最终确保各单元组合在一起后能够按既定意图协作运行，并确保增量的行为正确。

在系统集成测试中我们主要采用了代码审查，通过对核心源代码的阅读，发现代码中存在的诸如代码格式、逻辑错误等问题；通过对数据流程的分析，编写测试用例，进行动态测试，发现功能上的错误。

### 2.7.1 集成测试概述

系统集成测试是“LSIC社区”大学生学习生活信息交流网站的最终测试，是建立在开发组程序员开发完毕以及开发组单元测试完毕基础之上的。

由于我们不具备压力测试、安全测试、性能测试、运维测试等测试条件，这里仅就系统集成测试中的业务功能测试做详细介绍。

**测试地点**：组员各自家中

**测试时间**：2020年5月6日

### 2.7.2 测试方案

1. 生活服务模块测试：

可以实现代取快递、二手交易、租住房屋、兼职、组团拼车/拼单/出游等等的功能，用户可以在相应的分区自由跳转、发帖、评论、回复、领取任务等。

1. 考研保研模块测试：

通这个模块为了考研保研的同学专门设置。这个模块的功能兼容了资源专区和交流专区的功能，在这个模块中，用户可以发帖回帖、删帖、下载资源。

1. 资源专区模块测试：

资源专区是为了上传下载资源专门设立的模块。用户一样可以在这个模块中进行发布资源，删除资源和下载资源。但是这个模块的发帖功能与其他不同，有着标题和资源的限制。

1. 交流模块测试：

这个模块专门为了方便校内交流而设立。用户可以在东大资料、东大报纸、活动/竞赛三个版区任意选择。在这个模块内可以浏览东大资料、浏览校园网上报纸、管理活动/竞赛信息、查看活动/竞赛信息。

1. 账号信息管理模块测试：

用户可以进行注册账号、修改个人信息、注册账号、找回密码等操作。

1. 网站维护测试：

设置管理员，管理员可以浏览网站所有信息、更新东大资料、删除违规信息、更新校园报纸、管理账号信息等。

## 2.8 系统测试小结

1983年IEEE提出的软件工程标准术语中给软件测试下的定义是：“使用人工或自动手段来运行或测定某个系统的过程，其目的在于检验它是否满足规定的需求或是弄清预期结果与实际结果之间的差别”。

软件测试是保证软件质量和提高软件可靠性的重要手段。随着软件程序量和复杂度不断地增加，人们对软件质量的要求也在不断提高，软件测试在软件开发过程中占据的位置也变的越来越重要了，同时软件测试的工作量也显得越加艰巨。

随着软件规模和复杂度不断地增加，软件中的错误使得软件开发在成本、进度和质量上失控的可能性增大，导致了软件危机的出现。而软件测试出现可以使程序中的错误密度达到尽可能低的程度，是保证软件质量和提高软件可靠性的重要手段，是解决软件危机的重要途径。在这样的背景下，学习软件测试的方法就显得越来越重要了。

通过本次《软件工程》课程的学习，并通过本次大作业的切身实践，我们发现撰写详细的测试用例很重要。因为测试用例将指导我们测试工作的进行，可以有效而快捷的指导我们有目的的测试软件系统。

通过撰写黑盒测试用例以及白盒测试用例等，并严格按照测试用例来实施我们从测试使我们有效的发现程序了中的缺陷，为开发出能够满足用户的需求的系统提供了保障。

但是，在真正实施测试的时候，我们却发现软件测试是一件很繁琐的事情，往往需要花费大量的时间来进行黑盒测试和白盒测试，严重耽误了我们正常的开发进度。所以通过上网查询资料，我们发现其实已经有一些免费的软件测试工具能够一定程度上减轻软件测试人员的压力了，所以这里还是希望以后能有越来越多、越来越好用的测试工具出现，提高测试的效率和质量，从而节省软件开发的成本、提高软件的质量。

# 3 心得体会

教材中给出的软件工程的定义是：以质量为核心，为了经济地开发满足客户需求的软件而研究、建立和应用的系统化的、有规则的、可度量的和可控制的工程原则、方法，涉及软件过程、项目管理、开发方法、开发工具，甚至企业文化等各个方面。

在本次的软件工程大作业中，本团队完整地经历了选题、软件开发计划制定、用户需求分析、实施与测试的过程，每一步都完整地完成，最终总体上实现了最初的选题。在软件开发过程中，完成了大量图表的绘制，如甘特图、用例图、活动图、类图、功能层次图等，团队的各个成员都锻炼了本门课程的相关图表绘制。除此之外，在本次软件开发过程中，也涉及了大量本门课程的概念，如过程模型、需求分析方法、软件设计中的分治、抽象、耦合、黑盒测试、白盒测试等，在开发过程中，本团队成员在多次探讨中进一步加深了理解。

关于本团队采用的过程模型，是RUP模型、Scrum敏捷方法、XP敏捷方法的结合，我们以RUP模型为基础，结合Scrum项目管理框架，并加入XP核心实践，将项目的生命周期分为四个阶段，分别对应四次团队作业。在软件开发计划中，我们确定了采用的过程模型；在需求分析中，我们除了使用用例、对象-关系建模等需求分析方法外，还采用了Scrum开发模型中的用户故事，通过用户故事的编写，较迅速地明确了用户需求，并通过故事点估计了各个功能的复杂度；在系统设计中，除了系统设计、类设计、数据库设计、界面设计外，我们还加入了基于Scrum的迭代开发计划，用SR号、AR号为各个功能的开发编号，并制定迭代开发计划；在实施与测试中，我们吸取XP敏捷开发的经验，结对进行开发测试，并通过迭代的方法逐步进行网站开发，大大提高了软件开发的效率。

对于我们而言，这是一次从未有过且难忘的开发经历，是我们第一次全程线上进行团队开发，也是我们第一次进行网站的开发。然而，通过积极有效的线上文字探讨与语音开会，我们成功地完成了这次开发，团队的成员之间也建立了更默契的合作，在此我们感恩遇见彼此，同时也对最终完成任务而感到自豪。