CMS第一次设计报告

1. **小组成员**

前端：吴梦凝19301050

后端：吴梦凝19301050

设计：任城仪19301045

1. **概要设计**

本系统采用springboot+Mybatis+Spring security+thymeleaf+junit+Tomcat的方式进行实现。

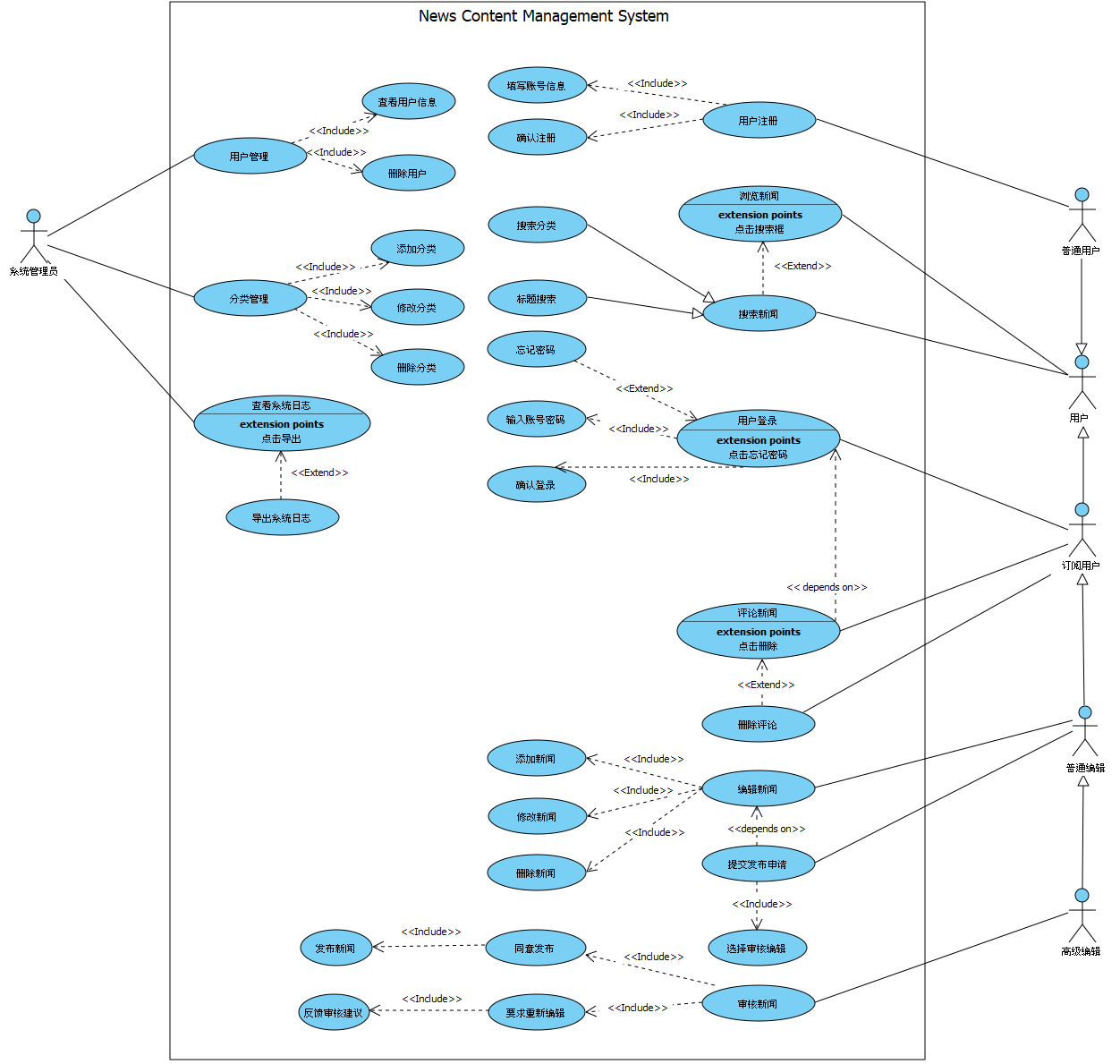
|  |  |
| --- | --- |
| Springboot | 用于实现后端数据处理和业务逻辑的功能； |
| Mybatis | 用于封装数据库，并实现springboot中的DAO层； |
| Thymeleaf | 用于实现前端界面； |
| Spring security | 用于对系统安全进行防护，包括授权和登录； |
| Junit | 用于完成单元测试； |
| Tomcat | 用于应用部署； |

1. **详细设计**
   1. **系统功能**

本新闻内容管理系统具有以下功能：

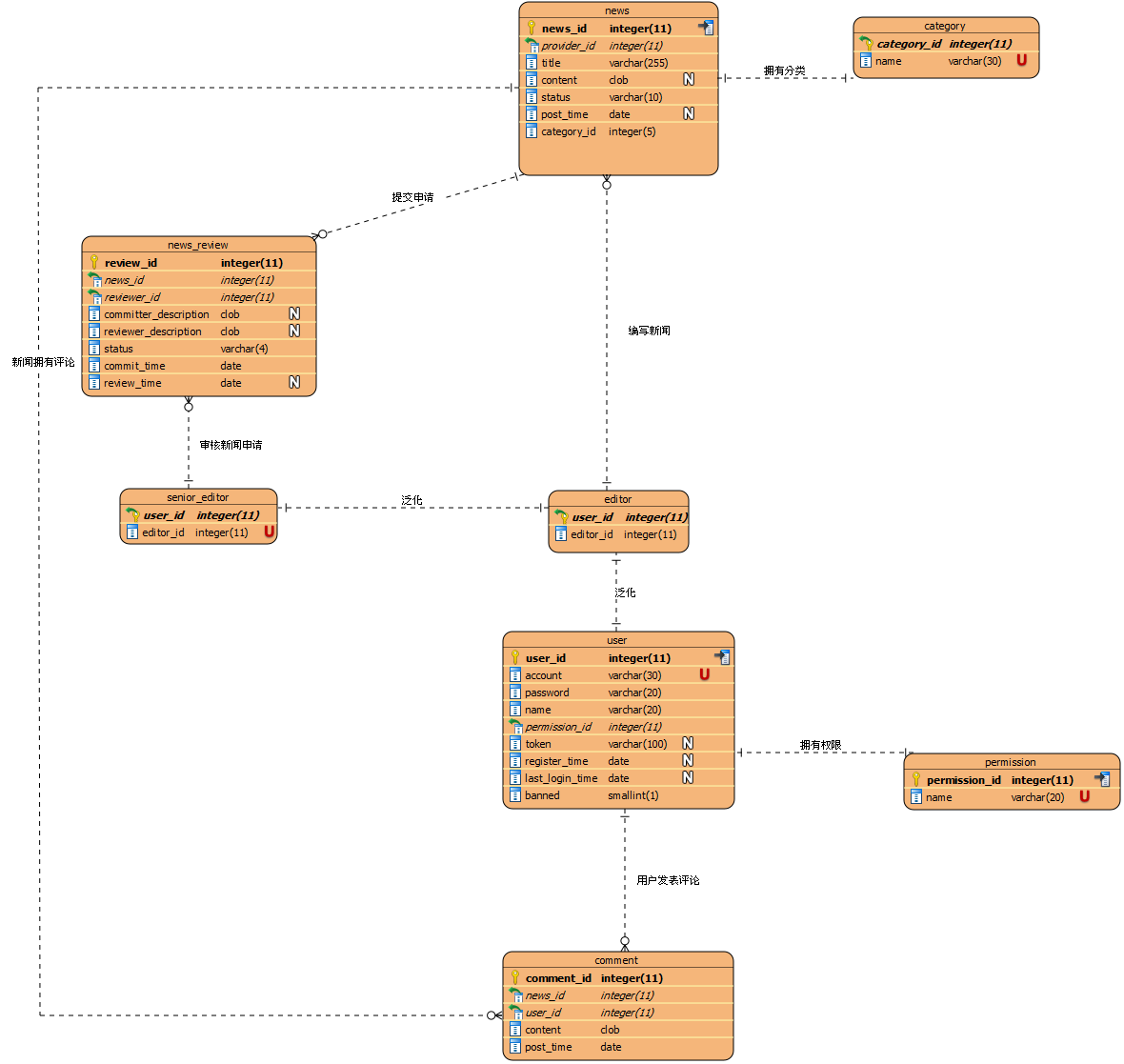
|  |  |
| --- | --- |
| 编号 | 功能描述 |
| 1 | 用户登录注册 |
| 2 | 用户阅览新闻 |
| 3 | 用户搜索新闻 |
| 4 | 用户添加评论信息 |
| 5 | 用户删除评论信息 |
| 6 | 编辑添加新闻 |
| 7 | 编辑提交发布申请 |
| 8 | 编辑管理自己发布的新闻 |
| 9 | 高级编辑审核新闻发布申请 |
| 10 | 高级编辑管理审核列表 |
| 11 | 管理员用户查看用户列表 |
| 12 | 管理员用户查看分类列表 |
| 13 | 管理员用户删除用户账号 |
| 14 | 管理员用户修改分类 |
| 15 | 管理员用户删除分类 |
| 16 | 管理员用户修改用户发言权限 |

具体用例图如下：



* 1. **数据库设计**

本系统主要包括用户，新闻，审核申请，权限和评论这几个实体，ER图如下：



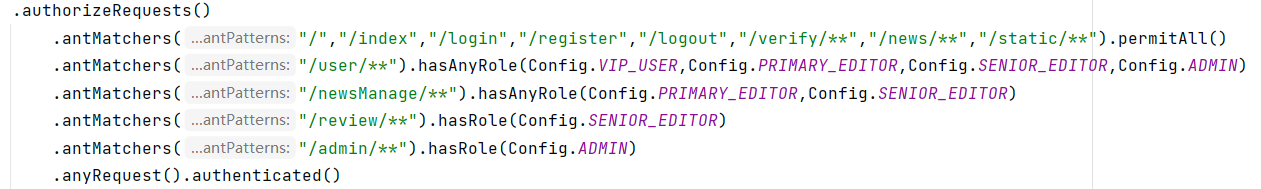
* 1. **Spring Security实现登录**

本系统通过security设置全局过滤（Filter）链，拦截对所有需要登录的功能界面，和其他界面设置的表单请求，并进行身份验证。此外，系统针对跨站点请求伪造（CSRF）设置了安全防护。

本系统通过设置自定义登录界面对用户提交的form表单进行验证，当验证成功时，系统将返回一个完全填充的身份验证实例，并跳转至首页。此外，当用户点击注销按钮时，系统会身份验证实例删除，并清除token，然后跳转至index界面。

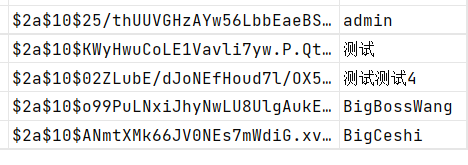
* 1. **Spring Security实现授权**

本系统包括5个不同的身份权限，普通用户（未登录），订阅用户（ROLE\_USER），普通编辑(ROLE\_EDITOR)，高级编辑(ROLE\_REVIEWER)和管理员用户(ROLE\_ADMIN)。这5个身份权限分别能够访问不同的功能，如订阅用户能够发布评论，编辑能够发布新闻，高级编辑能够审核新闻，管理员用户能够管理用户信息。具体权限如下：



* 1. **Spring Security实现密码加密**

本系统通过Security中的BCryptPasswordEncoder对用户登录，注册时输入的密码进行加密。因此，数据库中存放的是加密后的序列码，用户无法通过数据库中数据获取他人账号密码。下图为加密后数据库中的密码：

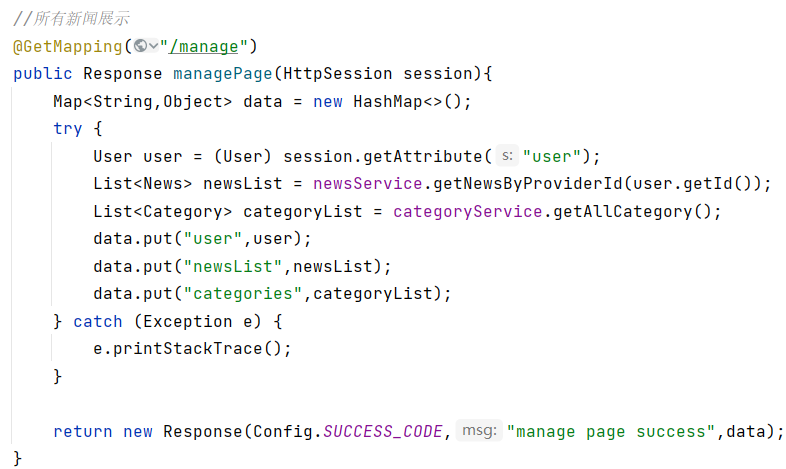


* 1. **Spring Security实现记住我**

本系统通过TokenBasedRememberMeService实现记住账号功能，当用户登录时选择记住我，系统会在登录后将安全token存入cookie。在token过期之前，用户能够随意关闭或重新打开界面，且不需要重新登录。

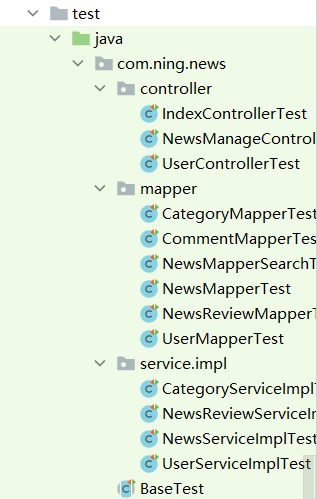
* 1. **HTTP session用于共享数据**

为了减少代码重复性，用户登录后，系统会将安全上下文中的身份验证对象存入Session。在其余前端界面中，thymeleaf将直接从session中取得相应的用户信息。



* 1. **Junit4 实现单元测试**

本系统使用Junit4完成了对Mapper，Service和Controller层的单元测试。具体的测试文件结构如下：



* 1. **自定义Error图片**

本系统通过对error错误码的重定向，完成自定义404，403，500界面设置。

1. **运行截图**

如下：

