****

软件开发实训

课程设计报告

题 目： **留言板**

学 院： 计算机与信息安全学院

专 业： 计算机科学与技术

学生姓名： 邹华梁

学 号： 14003103032

指导教师： 梁 海

2017年12月25日

目录

[第一章 详细设计 1](#_Toc502005232)

[1.1设计简介 1](#_Toc502005233)

[1.2 结构设计 1](#_Toc502005234)

[1.2.1 功能分解 1](#_Toc502005235)

[第二章 系统设计 2](#_Toc502005236)

[2.1 数据库设计 2](#_Toc502005237)

[2.1.1 message 表 2](#_Toc502005238)

[2.1.2 user 表 3](#_Toc502005239)

[2.1.3 role 表 3](#_Toc502005240)

[2.2 程序设计 4](#_Toc502005241)

[2.2.1 用户访问控制模块 4](#_Toc502005242)

[2.2.2 留言获取模块 6](#_Toc502005243)

[2.2.3 留言管理模块 8](#_Toc502005244)

[2.2.4 留言展示页面 9](#_Toc502005245)

[2.2.5 留言管理页面 9](#_Toc502005246)

[2.2.5 用户登陆页面 10](#_Toc502005247)

[2.2.6 用户注册页面 10](#_Toc502005248)

[第三章 功能测试 11](#_Toc502005249)

[第四章 总结 14](#_Toc502005250)

# 

# 第一章 详细设计

1.1设计简介

本软件分为前端和后端两大部分，其中前端部分有消息展示页面、消息管理页面、用户注册页面和用户登陆页面，后端部分分为控制模块，用户身份验证模块和数据库操作模块。本软件采用前后端分离的思想进行设计，前端通过 ajax 向后端请求所需数据，后端进行处理，并将数据打包成 json 返回给前端。

1.2 结构设计

1.2.1 功能分解

本软件分为前端和后端两大部分。

前端：

（1）留言展示页面

（2）留言管理页面

（3）用户注册页面

（4）用户登陆页面

后端：

（1）用户访问控制模块

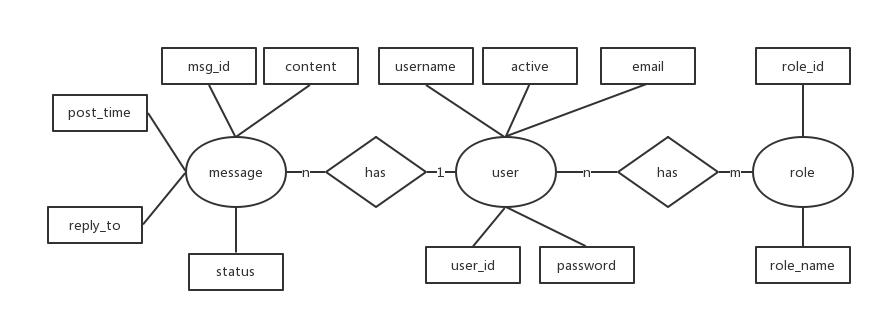
（2）留言获取模块

（3）留言管理模块

# 第二章 系统设计

2.1 数据库设计

数据库E-R图如图1.1所示：



数据库E-R图2.1

2.1.1 message 表

message表负责保存留言的基本信息。

message表的数据字典如表 1.1 所示：

|  |  |  |
| --- | --- | --- |
| 列名 | 类型 | 说明 |
| msg\_id | int | 主键，留言 id |
| content | varchar(255) | 留言内容 |
| post\_time | timestamp | 发送时间 |
| reply\_to | int | 所回复的留言的 msg\_id |
| status | int | 留言状态 |
| author\_id | int | 外键，作者 id |

message表的数据字典 表 1.1

2.1.2 user 表

user 表负责保存用户的基本信息。

user 表的数据字典如表 1.2 所示：

|  |  |  |
| --- | --- | --- |
| 列名 | 类型 | 说明 |
| user\_id | int | 主键，用户 id |
| username | varchar(255) | 用户名 |
| email | varchar(255) | 邮箱 |
| password | varchar(255) | 密码 |
| active | int | 用户状态 |

user表的数据字典 表 1.2

2.1.3 role 表

role 表负责保存用户的基本信息。

role 表的数据字典如表 1.3 所示：

|  |  |  |
| --- | --- | --- |
| 列名 | 类型 | 说明 |
| role\_id | int | 主键，角色 id |
| role | varchar(255) | 角色名 |

role表的数据字典 表 1.3

由于 user 与 role 是多对多关系，因此需要建立表 user\_role 来维护两者之间的关系。

2.2 程序设计

后端：

2.2.1 用户访问控制模块

用户访问控制模块负责对用户的身份进行验证，提供用户注册与用户登入与登出功能，并将未登录的用户引导至登陆页面。此模块的相关程序流程图如图2.2、图2.3和图2.4所示：

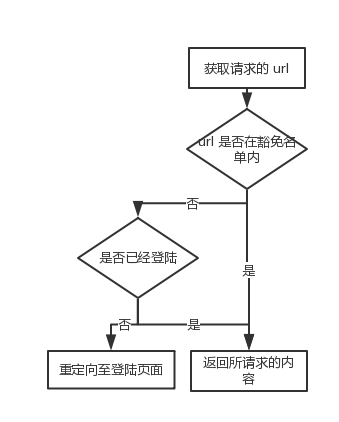


图2.2

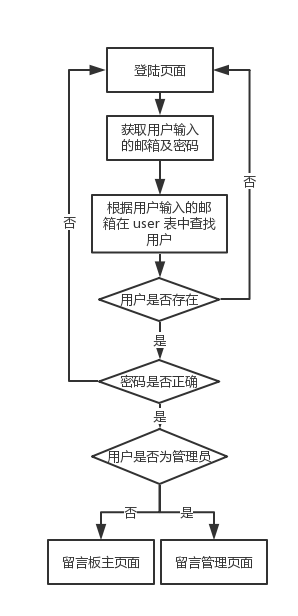


图2.3

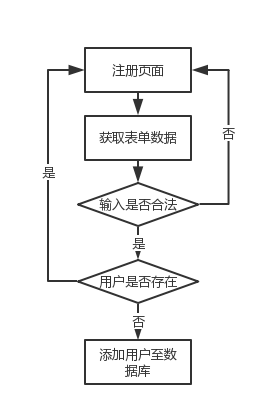


图2.4

2.2.2 留言获取模块

留言获取模块负责根据请求的内容来从数据库查询相应的内容，并将查询到的数据打包成相应的格式，返回给前端。此模块的相关程序流程图如图2.5和图2.6所示：

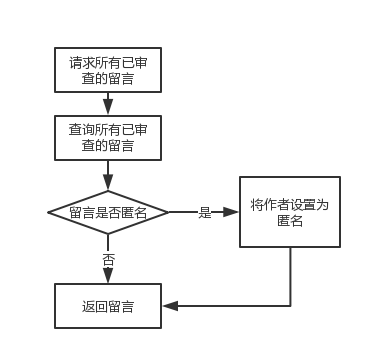


图2.5

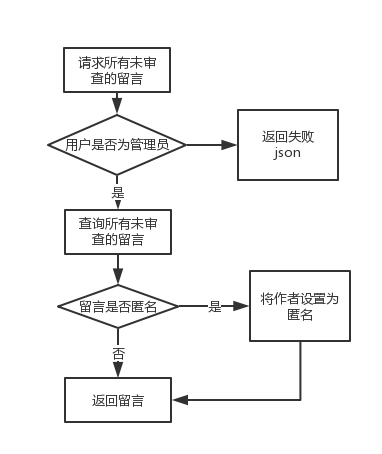


图2.6

2.2.3 留言管理模块

留言管理模块负责留言的审核以及留言的删除，只有通过审核的留言才会在留言板上展示，而留言删除不是直接将留言从数据库中删除，而是将留言的状态设置为删除状态。此模块的相关程序流程图如图2.7和图2.8所示：

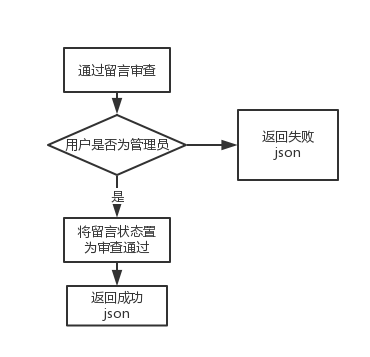


图2.7

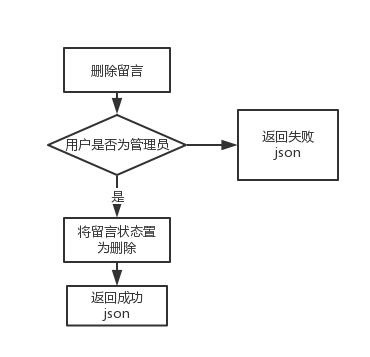
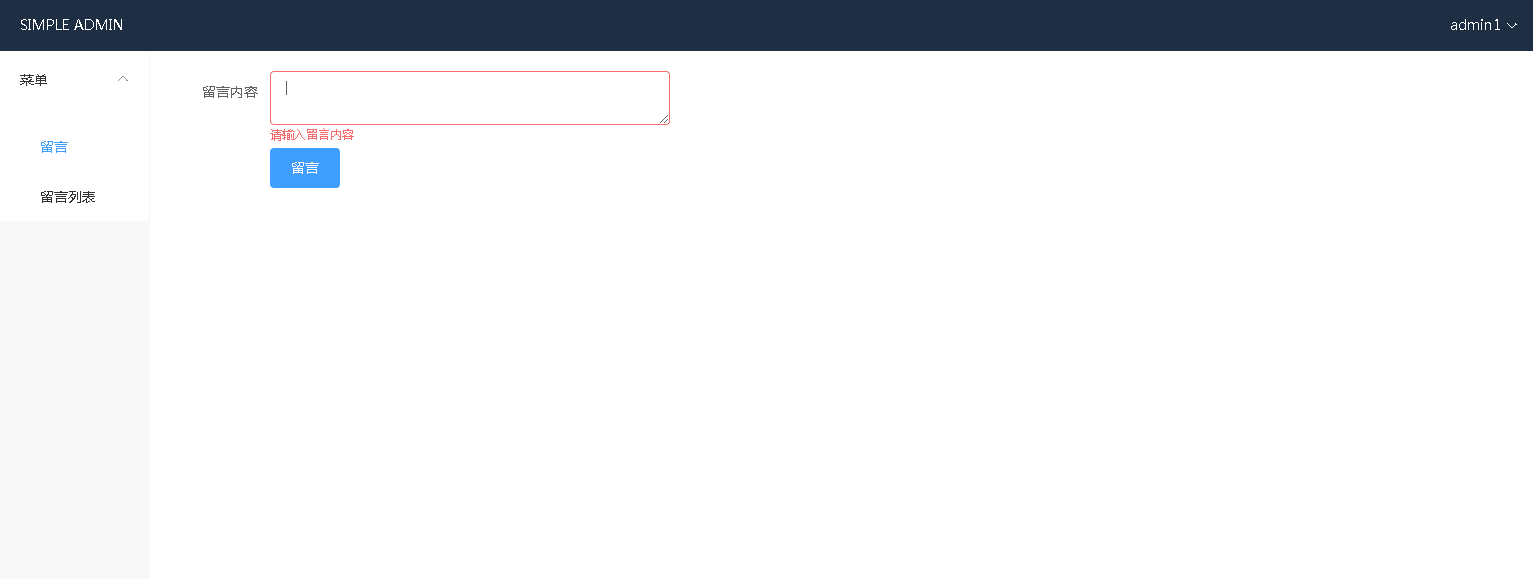


图2.8

前端：

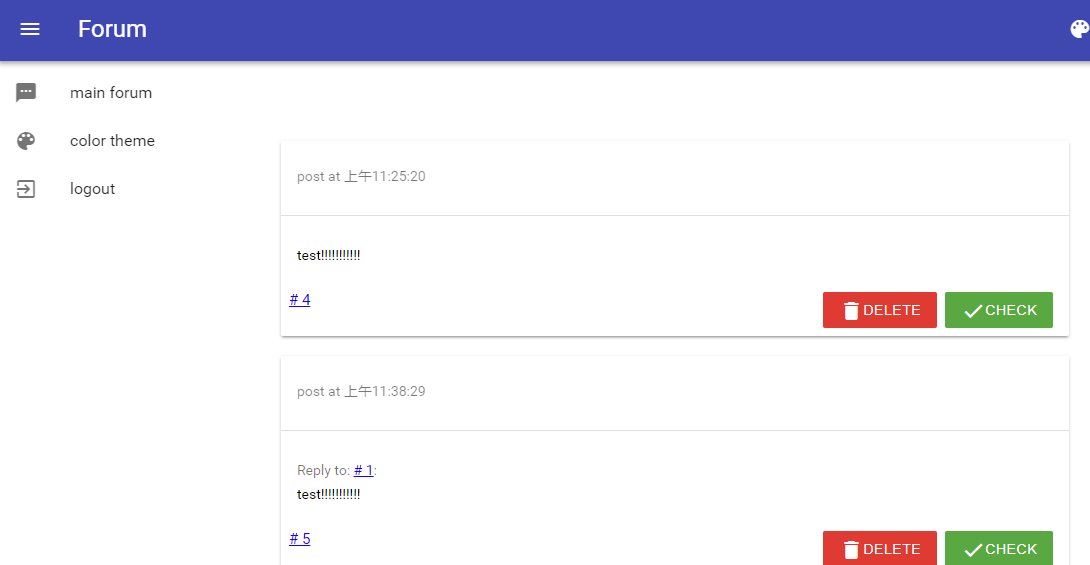
2.2.4 留言展示页面

留言展示页面负责向用户展示留言，并提供留言输入界面，方便用户进行留言。此页面的设计如图 2.9 所示：

留言展示页面 图 2.9

2.2.5 留言管理页面

留言管理页面负责为管理员提供留言审查与删除接口，方便管理员进行留言管理。此页面的设计如图 2.10 所示：



留言管理页面 图 2.10

2.2.5 用户登陆页面

用户登陆页面负责为用户登陆提供接口，方便用户进行登陆。此页面的设计如图 2.11 所示：



用户登陆页面 图 2.11

2.2.6 用户注册页面

用户注册页面负责为用户登陆提供接口，方便用户进行注册。此页面的设计如图 2.12 所示：



用户注册页面 图 2.12

# 第三章 功能测试

为了针对本系统的不同模块进行测试，本文设计了4个测试用例来对系统进行测试。

测试用例一用于测试用户访问控制模块的用户登陆功能，测试用例一详细设计如表3.1所示：

|  |  |
| --- | --- |
| 测试用例编号 | Test001 |
| 测试项目 | 用户登陆功能 |
| 预设条件 | 当前网络畅通 |
| 输入 | 正确的用户名与密码 |
| 操作步骤 | 点击“登陆”按钮 |
| 预期输出 | 正常登陆并跳转到首页 |

表3.1 测试用例一

测试用例二用于用户访问控制模块的用户登陆功能，测试用例二详细设计如表5.2所示：

|  |  |
| --- | --- |
| 测试用例编号 | Test002 |
| 测试项目 | 用户登陆功能 |
| 预设条件 | 当前网络畅通 |
| 输入 | 错误的用户名与密码 |
| 操作步骤 | 点击“登陆”按钮 |
| 预期输出 | 登陆失败并跳转到登陆页面 |

表5.2 测试用例二

测试用例三用于测试数据库操作模块的保存功能，测试用例三详细设计如表5.3所示：

|  |  |
| --- | --- |
| 测试用例编号 | Test003 |
| 测试项目 | 数据库操作模块的保存功能 |
| 预设条件 | 数据库连接正常 |
| 输入 | 留言bean |
| 操作步骤 | 输入留言并点击发送按钮 |
| 预期输出 | 留言成功保存，用户界面出现 success 弹窗 |

表5.3 测试用例三

测试用例四用于留言管理模块的留言删除功能，测试用例四详细设计如表5.4所示：

|  |  |
| --- | --- |
| 测试用例编号 | Test004 |
| 测试项目 | 留言管理模块的留言删除功能 |
| 预设条件 | 数据库中有未审核的留言 |
| 输入 | 未审核的留言中的一条 |
| 操作步骤 | 点击 delete 按钮 |
| 预期输出 | 留言成功置为删除，用户界面中该留言消失。 |

表5.4 测试用例四

# 第四章 总结

通过这次课程设计，我学会了版本控制系统 git 的使用方法，了解了前后端分离的设计方法，并通过在网上查找相关资料，按照前后端分离的设计方法，设计了完成留言板所需的 restful api，这使我对软件开发的了解更加深入了。