## 论文题目

姓名：魏心宇 学号：2017301550063 班级：计科8班

摘要

本项目实现了一个简易的社团平台管理系统，主要功能有多角色登录（管理员和普通用户）、管理员角色下的社团人事管理、公告管理、博客管理、社团财务管理和普通用户角色下的发布博客、查阅博客、查阅公告等。

选题背景：web开发可以说是一个程序员非常重要的一项技能，也是在工作中最常用到的技术之一，所以学习web开发是十分重要的。同时，我自己是学校微软俱乐部的一名成员，该社团的主要目的就是促进各种技术的学习交流；该项目一方面作为社团的简易管理平台，另一方面也可以作为一个框架项目供其它同学学习和进一步完善、填充内容。因此开发一个具有基础管理系统功能、文件结构清晰、便于后续开发和完善的web项目是一个不错的选题。

虽然web开发从原理上说是很清晰、简单的：前端负责展示给用户、向后端发送请求；后端负责核心业务逻辑、数据的持久存储等。然而在实际开发中，还是遇到了不少技术问题和难点。一开始，我想进行完全的前后端分离：前端采用Vue框架和一些主流的页面组件库进行开发；后端采用asp.net core进行api的发送，使用EntityFramework作为ORM映射框架，并采用非关系数据库MongoDb存储数据。但在开发过程中却屡屡遇到困难，究其原因，完全的前后端分离更适合拥有成熟框架和结构的javaEE（如spring boot），而asp.net core是一个相对较新的框架，这就导致了它在与Vue前端进行交互时屡屡出现各种意想不到的bug。所以最后还是采用了asp.net mvc作为后端框架、与asp.net兼容更好的sql server作为持久层。同时，要实现一个合理、易用的web页面，良好的逻辑结构和简单易用的前端页面是十分重要的。这里我使用了Vue.js作为前端框架，采用了layui作为页面控件库，并参考了指导老师的意见进行了页面布局与交互。

关键词：Asp.net、Vue.js、Layui

1. 背景和问题

本软件为《MSC社团管理系统》。

选题背景：web开发可以说是一个程序员非常重要的一项技能，也是在工作中最常用到的技术之一，所以学习web开发是十分重要的。同时，我自己是学校微软俱乐部的一名成员，该社团的主要目的就是促进各种技术的学习交流；该项目一方面作为社团的简易管理平台，另一方面也可以作为一个框架项目供其它同学学习和进一步完善、填充内容。因此开发一个具有基础管理系统功能、文件结构清晰、便于后续开发和完善的web项目是一个不错的选题。

我负责前后端全部功能开发。具体而言：

首要功能是实现多角色登录，即管理员与普通用户两个角色。两个角色的访问权限不同、前端界面也有所不同。

|  |  |  |
| --- | --- | --- |
| 功能\角色 | 管理员 | 普通用户 |
| 登录平台 | √ | √ |
| 增加/删除社团成员 | √ | × |
| 查看公告 | √ | √ |
| 发布/删除公告 | √ | × |
| 查看博客 | √ | √ |
| 发布博客 | √ | √ |
| 编辑社团财务信息 | √ | × |

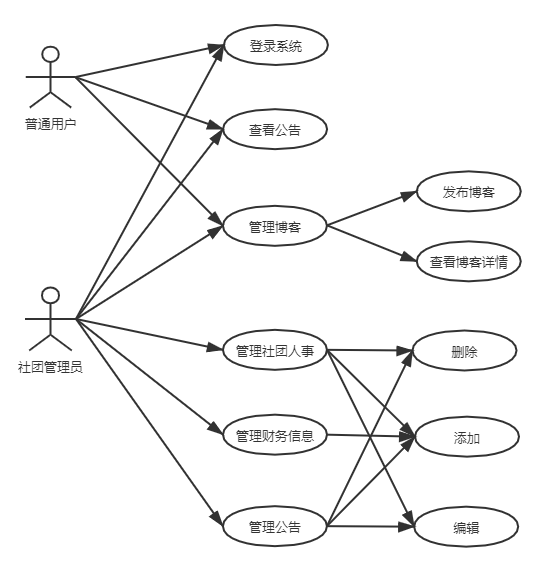
第一个问题是：如何在尽可能保证减少重复代码的同时，实现不同角色看到的界面不同？这里我采用的做法是：在加载页面之前判断登陆用户的角色，然后选择性地加载控件。

二是，如何绘制一个简洁、美观的前端界面？这里我参考了老师的意见，使用了layui控件库进行布局。具体效果请参考后文。

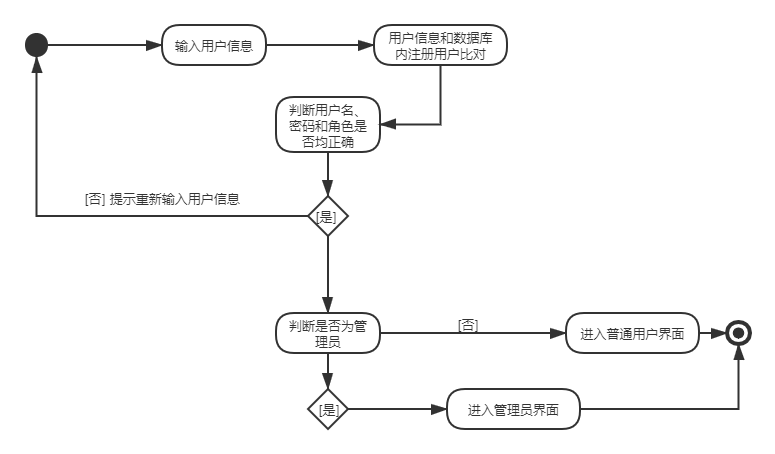
1. 需求分析

通过对上述功能的分析，我们不难根据需求画系统出用例图和功能流程图。

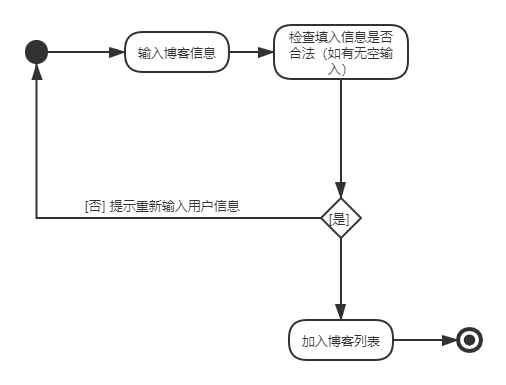
系统用例图：



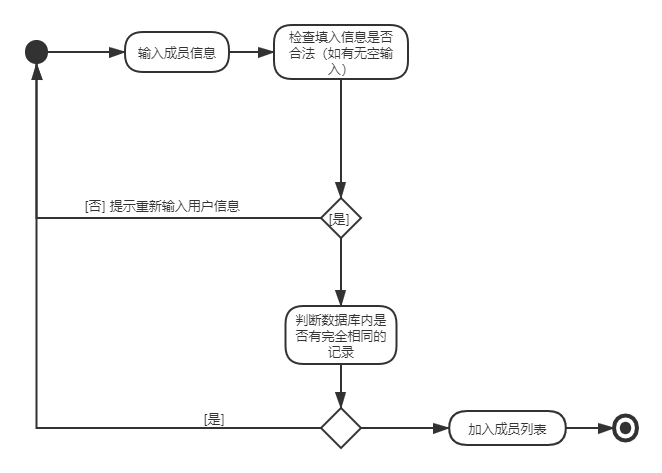
登录流程图：



添加博客流程图：



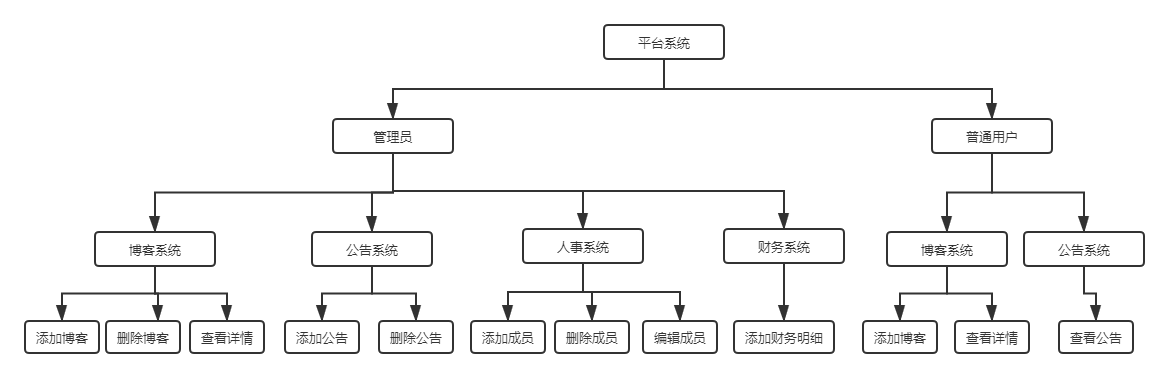
添加社团成员流程图：



以上为几个主要功能的需求分析。

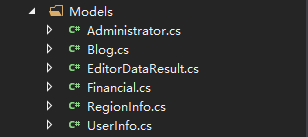
1. 模块设计

首先我们对管理系统的功能进行简要图示（功能组织图）：



为了使整体架构更为清晰、不同层级的逻辑结构更加独立，我将项目主体类分为三层：

Models层：用于封装实体类；



Administrator:管理员/普通用户实体

Blog：博客实体

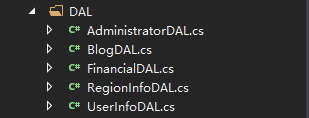
EditorDataResult：用于封装添加博客提示信息

Financial：财务信息实体

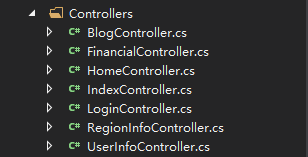
RegionInfo：公告实体

UserInfo：社团成员实体

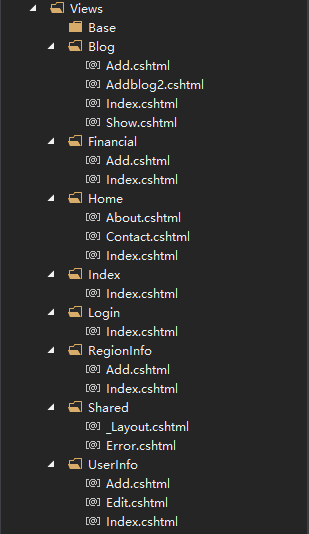
DAL层：用于封装核心业务逻辑，sql语句也在这一层中；



Controller层：用于接收前端的请求，并调用DAL层中的核心业务逻辑；

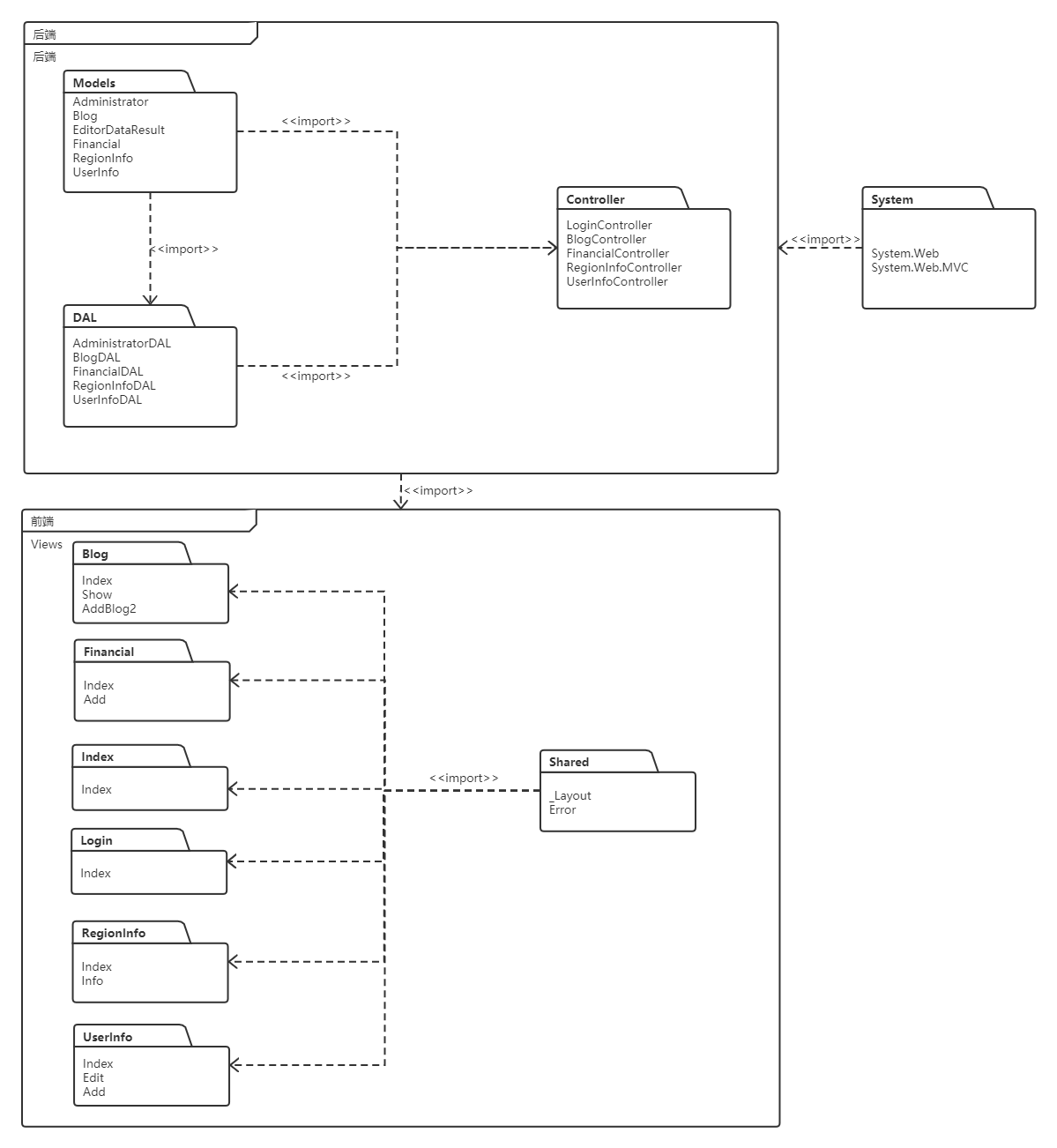


同时，页面部分放入views包中：

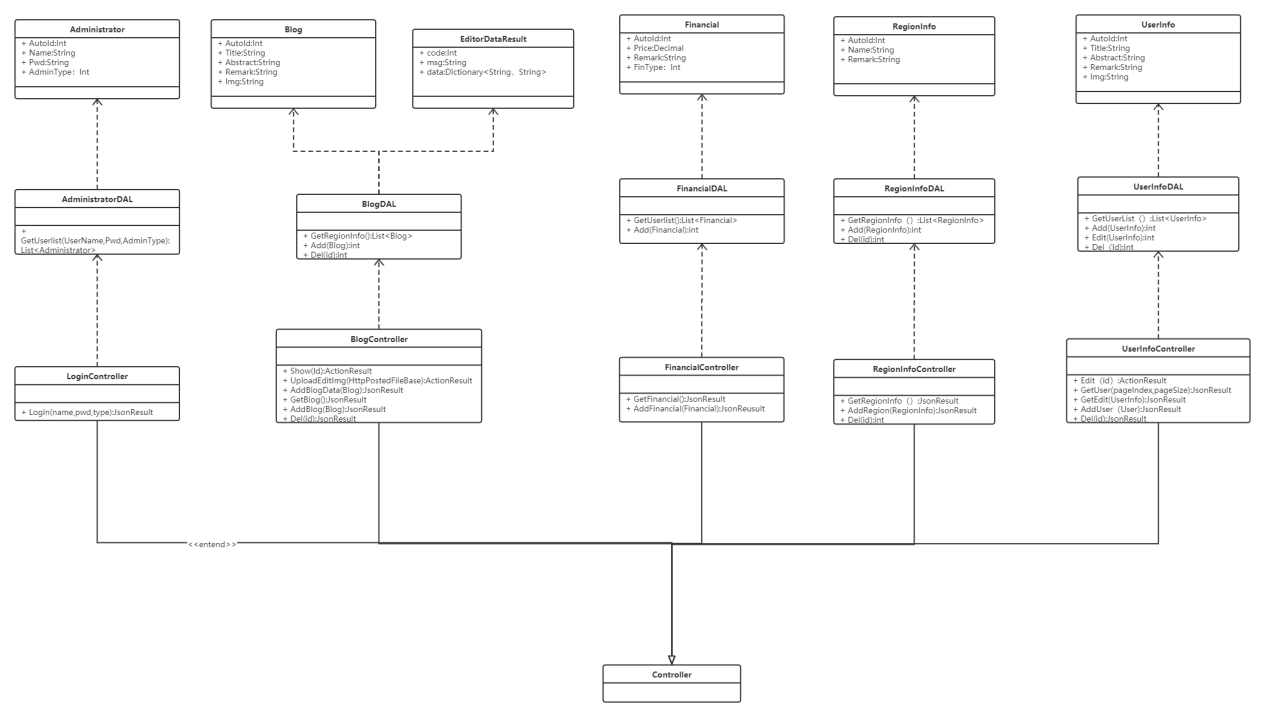


其中，shared包下为每个页面都共有的组件：导航栏、报错信息。故单独摘出作为组件供其他页面调用。

上述文件结构用包图表述文件结构则为：



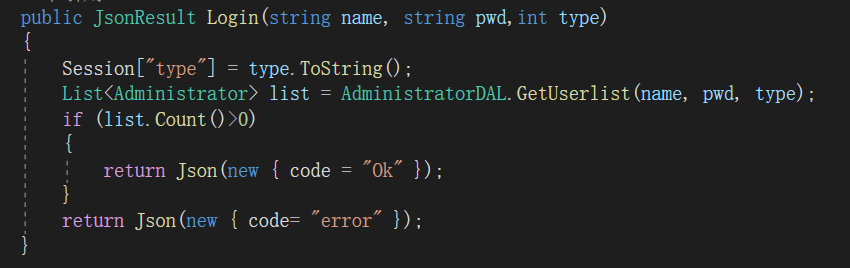
同时根据项目需求，我们不难得到类图：



1. 关键技术

问题1：如何在尽可能保证代码重用的前提下，使得不同角色登录后看到不同的界面呢？这里我采用的方法是：判断用户的角色，然后有选择地加载页面。这样就确保了不会有重复的页面代码。

相关代码：



问题2：如何在博客内容中插入图片？要以什么形式存储呢？

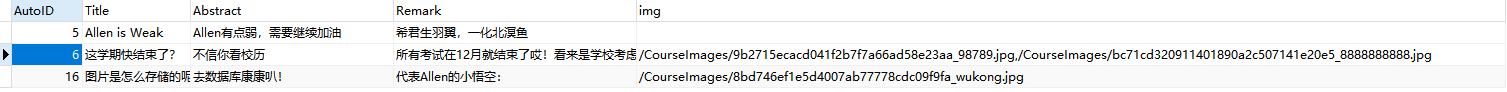
我使用了layui空间中的富文本框，可以实现点击按钮上传图片的功能；在数据库中，图片以String存放路径；而图片文件的实体存储在服务器上的特定目录下。举例：



在博客列表，我们进入《图片是怎么存储的呢》详情页面：

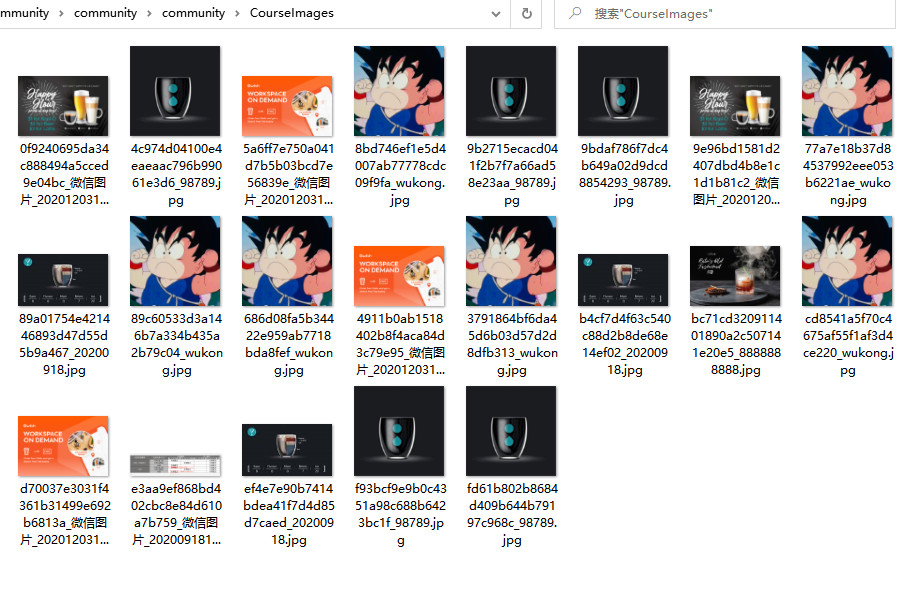


在数据库中：



Img的存储形式是一个字符串，记录的是图片文件的实际存储位置。

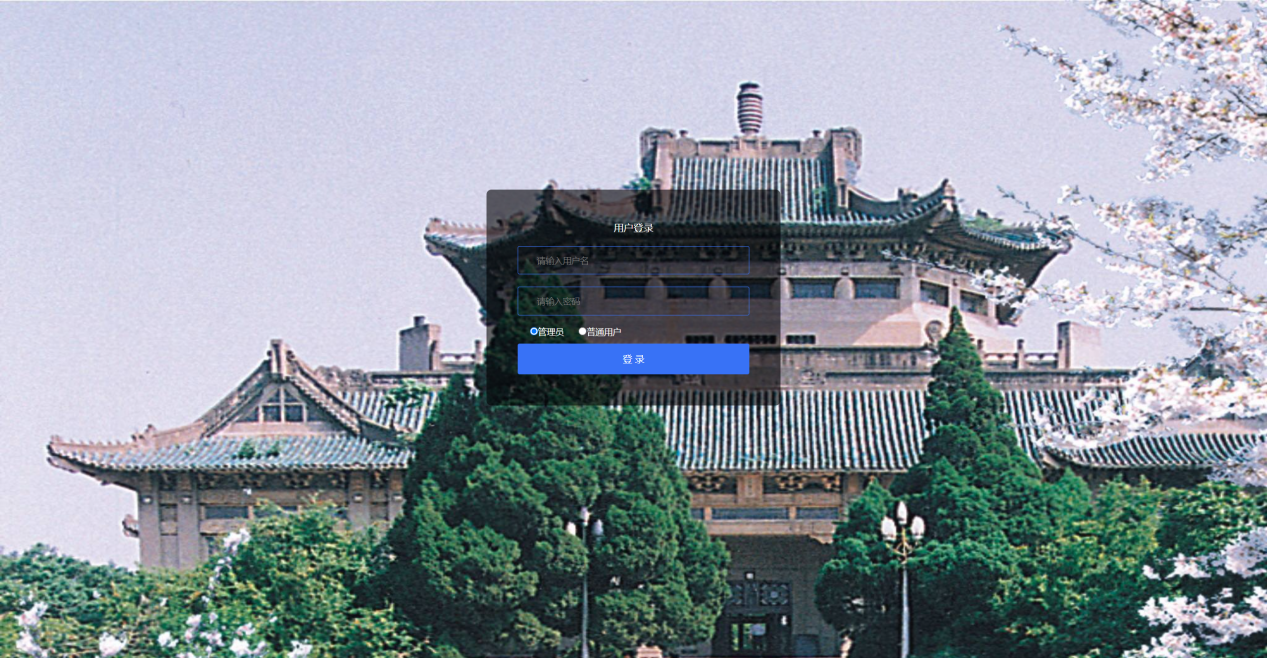
在项目目录中：



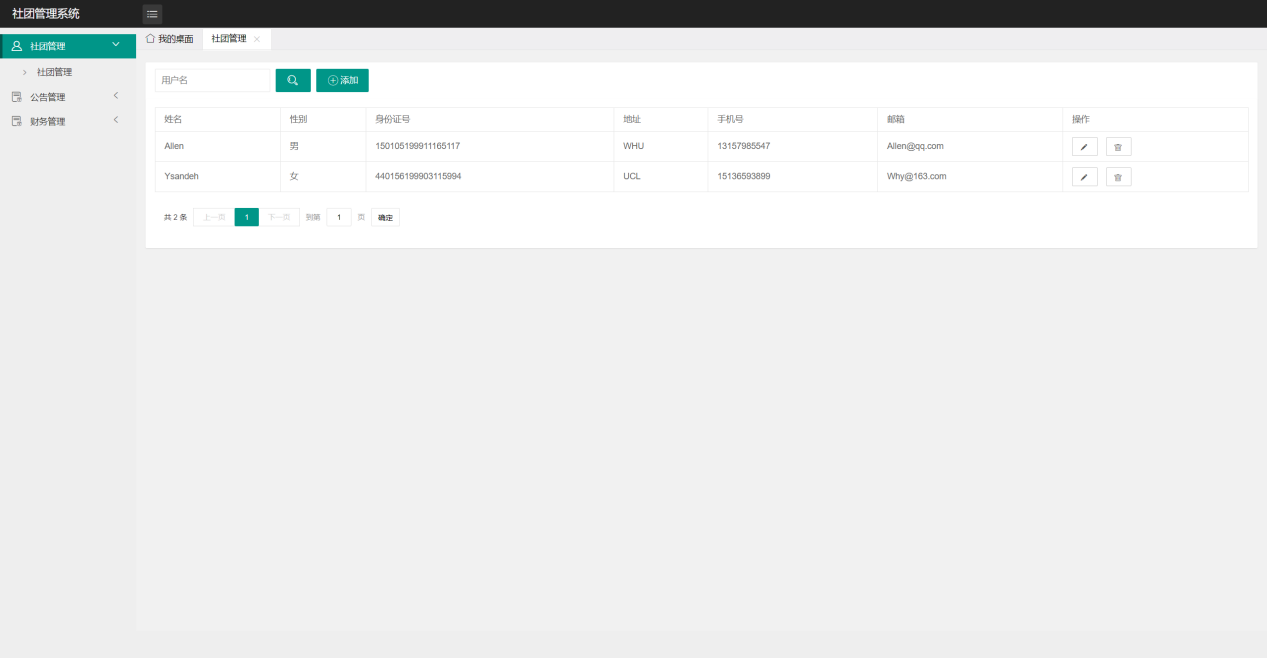
图片文件确实存储在了这里（请无视我测试都用同一张图片……）！

1. 实现效果

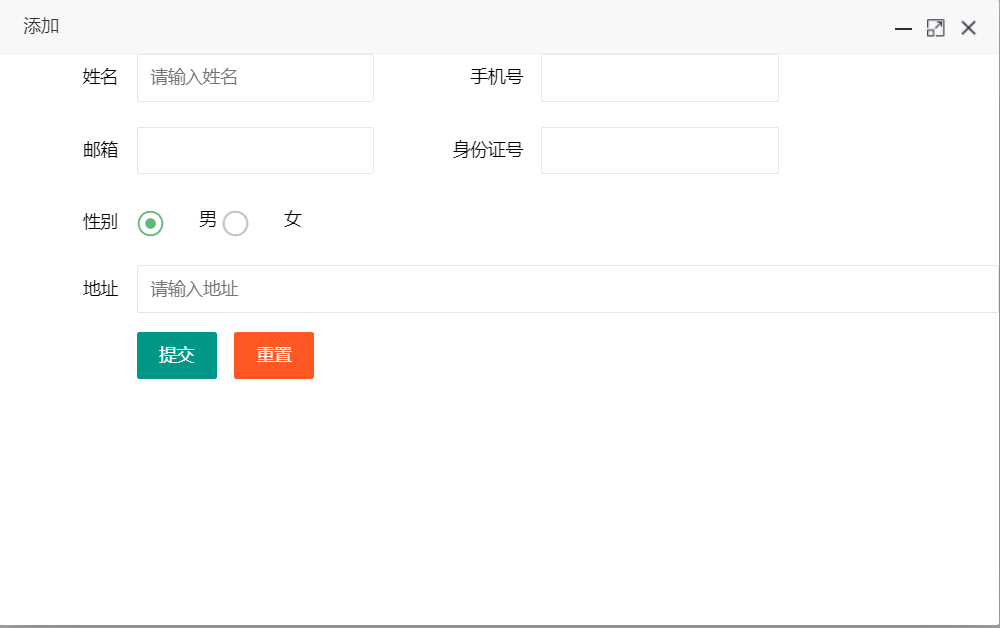
登陆界面：



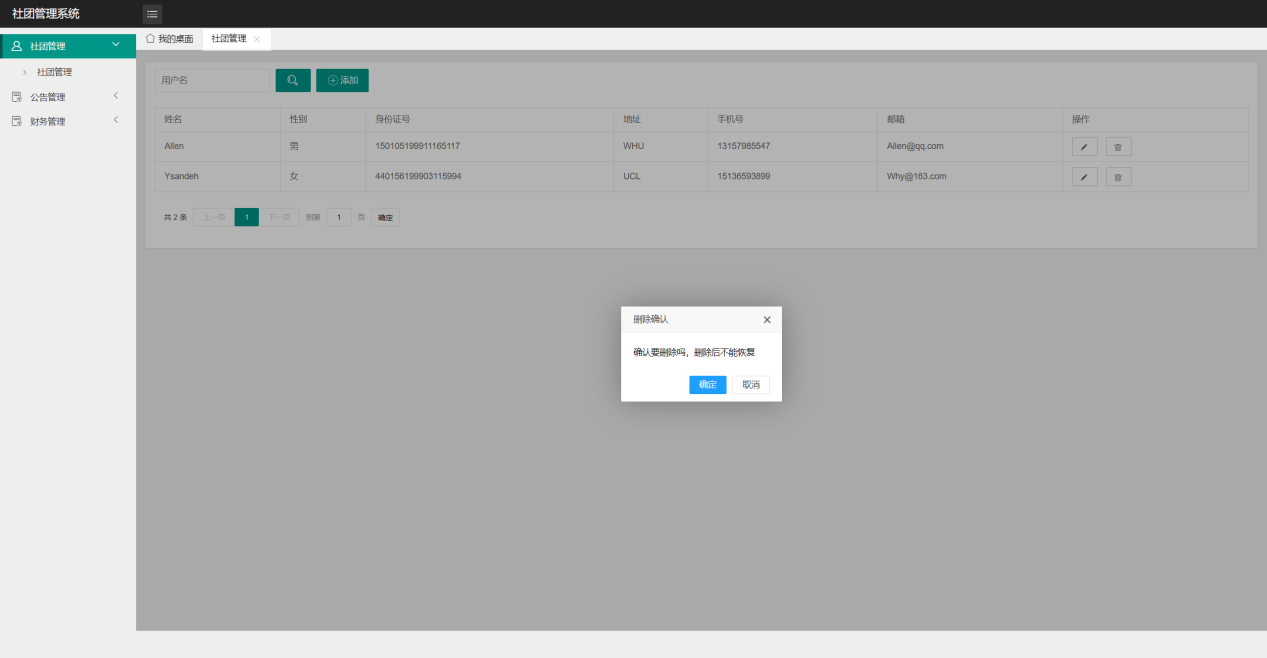
管理员人事界面（管理员的左侧菜单栏有全部功能）：



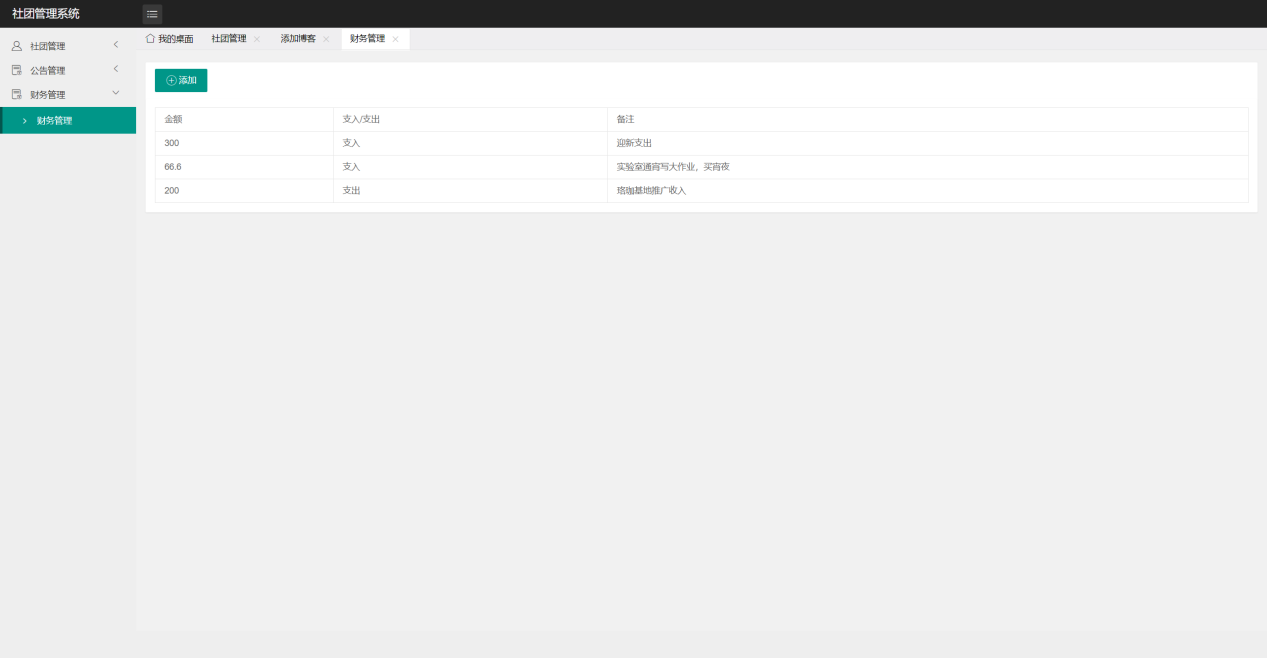
管理员添加成员：



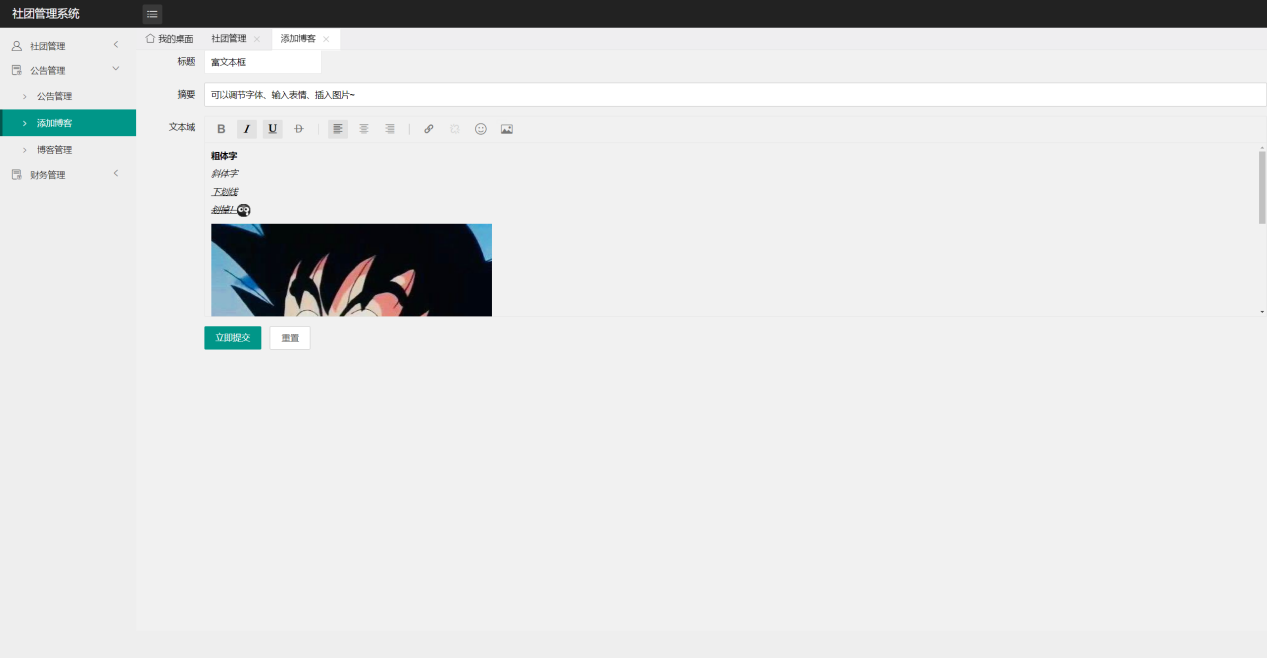
管理员删除成员：



管理员财务信息界面：



添加博客（使用富文本框）：



普通用户的菜单栏（只有有限的功能）：



1. 结论

最终，本项目完成了基本的社团管理系统功能，实现了前述所说的全部要求。在开发过程中，我也有了如下收获：

1. 对web开发有了更深刻的了解，对于前后端运行的原理、如何交互有了实践性的认识。网页开发是程序员较为重要的一项技能、也是工作中运用最多的技能，这次大作业的经历对于今后的学习乃至工作都受益匪浅；
2. 对大型项目的分层架构有了实践性的认识。之前也学习过springboot框架，对于架构分层有初步的学习了解；在asp.net环境下亲手实现Model-View-Controller的结构，使我对“分层”的概念有了更具体的认识；
3. 认识到了前端技术的重要性、锻炼了前端技术。具体开发了一个web项目后我发现：网页开发中前端开发比后端开发更难、也更重要。优秀的交互界面是一个良好网页的基础。而我在之前的学习中缺乏前端知识的学习，在开发过程中恶补了许多相关知识，初步掌握了Vue框架、Layui、Elementui(本质上其实基本相同)等控件库，前端技术有所提升；
4. 学习了当下较为流行的非关系数据库MongoDb（虽然最后没有使用MongoDb），学习了navicat系列可视化数据库操作平台，强化巩固了数据库相关知识；

回顾整个大作业开发过程，我也深刻认识到了自己的知识水平不足和开发水平稚嫩，在之前学习中缺乏实际项目的开发经验，很多知识的学习只是“纸上谈兵”，使得我在开发过程中屡屡遇到各种困难，经常需要查询大量资料。

这次课程不仅使我学习到了许多dotnet的知识，还提升了我对实际项目开发的兴趣；授课老师非常认真负责，也是我遇到的唯一一个愿意花时间在课程群帮助学生解答问题的老师，备受感动。选这门课之前，我还对web开发一窍不通、对项目架构等知识也一知半解；现在已经对web开发有了较为全面的认识、可以亲手搭建一个简单的网页，也对dotnet中常见的技术、知识有了了解，同时也对软件工程中的许多概念有了有了更实践性的认知（如为何要分层设计、为何要前后端分离等）。综上所述，这门课程对我今后的学习和工作都有巨大帮助，实在获益良多！