**目 录**

[**1 绪 论** 1](#_Toc4284)

[1.1 研究的背景及意义 1](#_Toc6975)

[1.1.1 选题的背景 1](#_Toc8727)

[1.1.2 国内外研究现状 1](#_Toc4120)

[1.1.3 研究的意义 2](#_Toc7525)

[1.2 系统目标 2](#_Toc4800)

[**2 需求分析** 4](#_Toc12631)

[2.1 业务需求 4](#_Toc5708)

[2.1.1主要业务流程 4](#_Toc12081)

[2.2 功能需求 8](#_Toc12137)

[2.2.1 角色分析 8](#_Toc10105)

[2.2.2 业务功能 8](#_Toc2255)

[2.3 非功能需求 23](#_Toc23705)

[2.3.1环境需求 23](#_Toc7726)

[2.3.2 性能需求 24](#_Toc22192)

[2.3.3 安全需求 24](#_Toc26394)

[**3 总体设计** 26](#_Toc13057)

[3.1系统设计的原则 26](#_Toc5896)

[3.2 系统体系结构设计 26](#_Toc5675)

[3.3 系统功能结构设计 28](#_Toc20432)

[**4 数据库设计** 30](#_Toc5034)

[4.1 概念结构设计 30](#_Toc29678)

[4.1.1 设计思路 30](#_Toc5402)

[4.1.2 E-R图 30](#_Toc23388)

[4.1.3实体属性设计 31](#_Toc31792)

[4.2 逻辑结构设计 33](#_Toc11289)

[4.2.1 设计思路 33](#_Toc23970)

[4.2.2 逻辑模型 34](#_Toc23527)

[4.3 物理结构设计 38](#_Toc25708)

[4.3.1 存取方式 38](#_Toc19725)

[4.3.2 存储结构 39](#_Toc10251)

[**5 界面设计** 43](#_Toc17953)

[5.1 界面关系图或工作流图 43](#_Toc21915)

[5.2 界面设计成果 44](#_Toc17702)

[5.2.1 主界面 44](#_Toc19212)

[5.2.2 子界面 45](#_Toc28795)

[**6 详细设计** 47](#_Toc31933)

[6.1 系统主要功能模块介绍 47](#_Toc15476)

[6.2功能模块设计 48](#_Toc328)

[6.2.1用户管理 48](#_Toc21142)

[6.2.2信息发布 52](#_Toc15420)

[6.2.3咨询反馈 56](#_Toc28617)

[6.2.4线下研修室 61](#_Toc10264)

[**7 编码** 67](#_Toc22779)

[7.1 代码实现与核心算法 67](#_Toc206)

[7.2 代码优化 70](#_Toc16911)

[**8 测试** 73](#_Toc23435)

[8.1 测试方案设计 73](#_Toc8460)

[8.1.1 测试策略 73](#_Toc23964)

[8.1.2 测试进度安排 74](#_Toc30687)

[8.1.3 测试资源 74](#_Toc5035)

[8.1.4 关键测试点 74](#_Toc12603)

[8.2 测试用例构建 75](#_Toc13220)

[8.2.1 测试用例编写约定 75](#_Toc20816)

[8.2.2 测试用例设计 76](#_Toc9986)

[8.2.3 关键测试用例 76](#_Toc377)

[8.2.4 测试用例维护 77](#_Toc26716)

[**9 总结与展望** 78](#_Toc30221)

[9.1 设计工作总结 78](#_Toc22109)

[9.2 未来工作展望 78](#_Toc26609)

[谢 辞 79](#_Toc28568)

[参考文献 80](#_Toc26539)

[附录A 外文翻译—原文部分 82](#_Toc17872)

[附录B 外文翻译—译文部分 86](#_Toc15863)

[附录C 软件使用说明书 89](#_Toc11437)

[附录D 主要源代码 91](#_Toc28217)

# 1 绪 论

## 1.1 研究的背景及意义

### 1.1.1 选题的背景

随着5G时代的到来,网速越来越快,人们对网络愈加依赖，高校紧跟时代步伐,对外交流逐渐增多。在这个信息高速传递的时代，高校应认清自身定位与作用，以紧跟时代的心态去拥抱新未来学生的互动需求，在高校平台上增强传播力、加大影响力和提高服务能力，高校转型势在必行。

通过充分整了解当下学生需求、完善网站功能、提高服务效率、提升服务质量、优化功能模块、添加特色栏目，在统一标准、统一规划、统一架构的集约化建设理念指导下，通过门户平台架构，以协同创新中心的业务需求为导向，以服务驱动和技术支撑为主线，打造高校的协同创新综合服务平台。

### 1.1.2 国内外研究现状

调研外国诸如美日欧等高校进行协同创新推进举措我们了解到,国外高校协同创新方向主要体现为:国外的高校将学习、研究、实践融为一体,政府进行鼓励与支持,私人科技中介公司为协同创新成果建立注入活力,为还未成熟的高校协同创新平台等提供技术或资金上的支持。我们吸取国外高校的经验，添加学生与老师参与协同创新的奖励机制、加强对高校发展协同创新的引导、加快推进高校协同创新平台建设将对我国高校协同创新发展发挥重要的促进作用。

步入信息化时代以来,全国各地的高校一直是未来各种技术人才的孵化园,协同创新对当地的经济发展、教育提升、科技创新起到了至关重要的作用。目前,随着万物互联、信息共享等现代高等教育理念的不断深入发展,以及各省市对人才日益激烈的渴求,地方高校必须进行改革,及时作出适当的调整,以保证教育的发展以及地方人才的需要。

2014年国务院出台《国务院关于加快发展现代职业教育的决定》(国发[2014]19号)以来,各省市高校都在积极探索转型发展路线并为此进行了大量的实践,获得了很多有用的成果。但是,在制约地方高校转型发展的外部影响因素还有很多,对地方高校转型发展的内部制约因素方面体现在大部分教师还保持着传统人才培养的观念,学校设施无法达到新发展的要求、不健全的人才培养机制等方面,对高校转型发展存在的问题及实施的方案还没有形成深入、全面的研究成果。

### 1.1.3 研究的意义

为了能更好地将资源进行整合并合理利用,避免资源浪费，提高学生以及教师协同创新的效率。本项目利用先进的互联网技术来建设高校协同创新综合服务平台，利用共享资源提升个人的综合能力与创新性，为高校师生的协同创新合作提供了更多选择，进而提升整体教育水平。

协同创新平台为高校学生进行协同创新提供了途径，降低了学生与教师之间交流合作的困难,为建设提供充足的人才资源和创新力量,对推进社会、经济和高等教育事业的发展,提高区域可持续发展竞争力等具有重要的战略意义。

## 1.2 系统目标

本系统应使用SSM技术架构，能够保留很多功能接口，以便于未来对系统功能的外拓或兼容。系统应采用B/S结构，在中心机房部署服务器、数据库服务器和应用系统，在未来进行分布式部署，以提高用户访问速度。用户只需要通过浏览器便能够访问网站进行操作，B/S架构为用户提供了极大的使用便利。本系统需要实现信息的发布子系统、咨询反馈子系统、线下研修室子系统、调查问卷子系统、技术论坛子系统、线上交流子系统，而本人负责用户管理、信息的发布子系统、咨询反馈子系统、线下研修室子系统这四个子系统的设计与实现。

1. 界面设计要求
2. 展示给用户的界面应该简洁大气，内容显示清晰明了，排版应该有一定的间隔，保证既不显得空洞又不会显得拥挤。
3. 颜色应该显得明艳，展示出大学生的朝气蓬勃，但应该保持一种主色调，每个页面主色调要统一，不能每个页面的主色调各不相同。网页字体要保持适中，标题要大而明显，关键部分要加粗，让人能够很快找到关键信息。页面跳转与加载时间不能超过5秒，对于页面内容过多的部分采用分页或流加载来减少数据传输时间。对主流浏览器(如：IE、火狐、chrome等)具有较好的兼容性，每条信息都要按照最新在上的原则显示，如有必要则在后面显示具体时间，发布信息如有引用或转载，则需要注明，若发现有未注明，可进行举报。
4. 布局要适当合理、层次分明，应该有头部导航栏、侧边导航栏、中间内容部分。头部最上面应该有网站的图标、文字介绍等；中部有合理的排版，丰富的内容，应该图文都有；底部要有跳转外链、二维码、详细介绍等内容。
5. 导航栏中的点击要明显，鼠标悬停能有具体的显示，对于一行中超出部分能以省略号表示。
6. 信息发布
7. 网站管理员定期发布一些最新的通知公告，比如国家的政策、国家级、省级项目活动、竞赛等，大学生查看这些信息后，如果有兴趣便可以私下进行组队或者报名参与这些活动竞赛。
8. 信息发布项须包含：公告标题、文号、稿源（作者）、网页标题、公告内容、发布时间（默认是当前时间）等，用户能够明确的看到发布的公告的详细信息。同时发布公告拥有以下功能：能够插入外链，并以特殊颜色显示出来，可以插入图片，设置居中、居左、居右等。
9. 能够通过公告标题、内容、作者、发布时间等进行快速检索，检索响应时间应该小于４秒。
10. 咨询反馈
11. 咨询反馈是面向问题的，主要目的是解决问题，在这里学生可以更加快速的了解一些自己想要知道的本校的一些信息，而不用去百度找本校的相关问题，如果有的问题无法解决，甚至可以请求当面解决。
12. 提供问答创建与搜索功能，能够对违规信息进行举报，管理员将在三日内处理该事项并将处理结果发送给举报用户，并对违规信息发布者进行警告或惩处。
13. 网上调查要求

允许师生在平台发布所需调查问卷，可方便快捷的收集整理问卷，得到反馈。可以自由填写调查问卷的内容，在截至时间后可以看到调查问卷的结果。

1. 线下研讨要求

沟通后，同校师生可在校内建设的研讨室内进行面对面的交流与合作，共同完成 某项实验活动或任务，能够在平台上自主预约或取消合作室房间。

# 2 需求分析

## 2.1 业务需求

1. 用户能够实现公告发布、查询人中心可以查看自己曾经发布的公告，并且能够将自己发布的公告进行删除。
2. 用户能够进行问题的填写、问题查询、举报违规等功能，管理员定期对提问与回答进行审核，对于违规的信息进行删除。
3. 实现对研修室信息的处理，如管理员进行创建研修室，修改研修室资料，用户查找研修室，进行预定，退定等操作。
4. 后台管理员能够管理网站的部分数据，管理员能够对问答信息信息、公告信息、研修室资源等信息进行管理与维护，并且管理员应及时处理用户所提交的举报。
5. 页面加载速度要快，采用HTTPS协议，尽量减少系统风险。
6. 系统能够应对一定量的请求吞吐，在出现问题时有相对应的解决手段。

### 2.1.1主要业务流程

1. 用户前台公告添加活动图如图2-1所示：

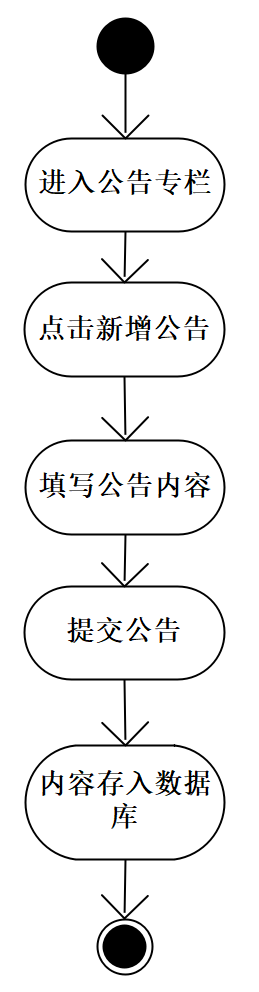


图2-1 公告添加活动图

1. 用户前台个人公告删除活动图如图2-2所示：

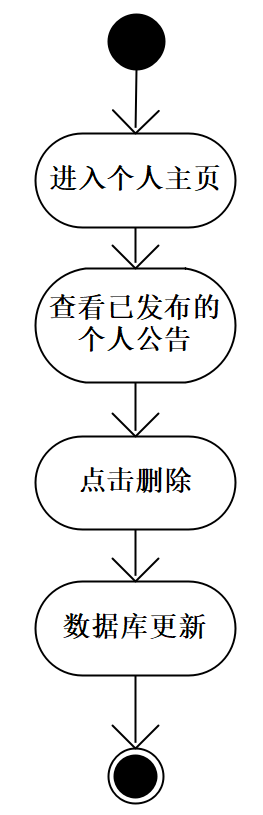


图2-2 网公告删除活动图

1. 用户前台问题查询活动图如图2-3所示：

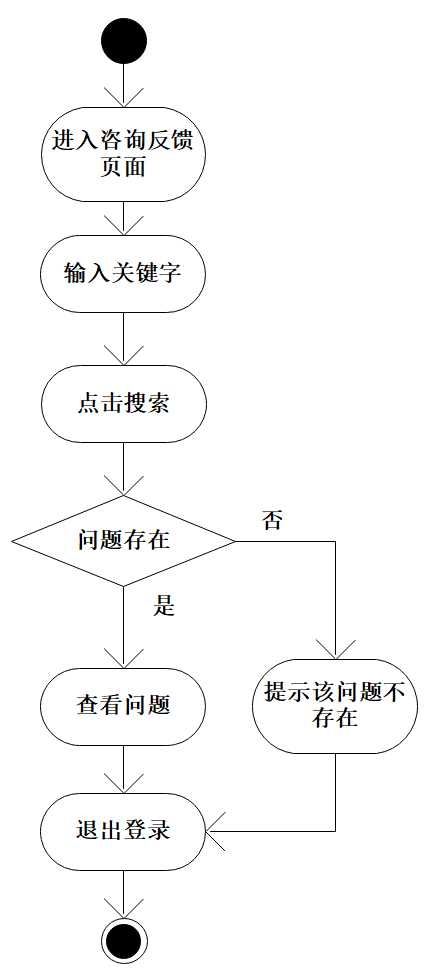


图2-3 问题查询活动图

1. 用户前台问题添加活动图如图2-4所示：

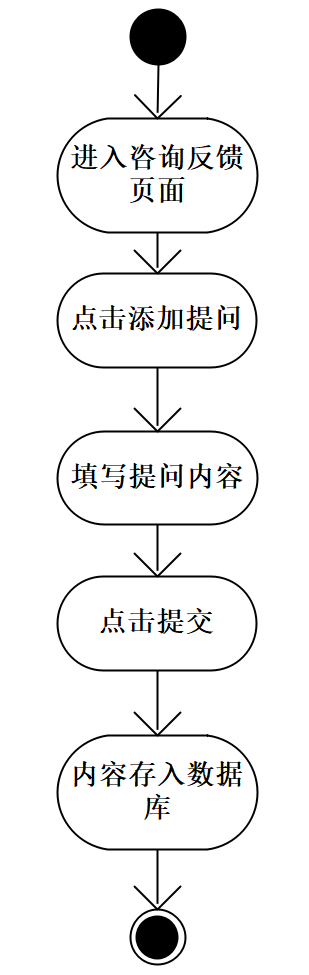


图2-4 问题添加活动图

1. 用户前台个人问题删除活动图如图2-5所示：

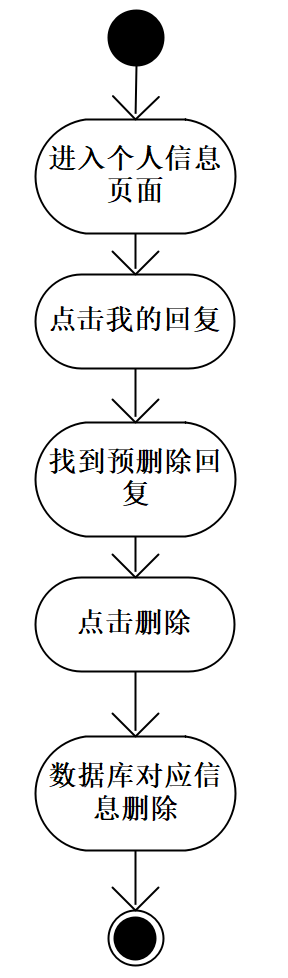


图2-5 问题删除活动图

1. 管理员后台研修室创建活动图如图2-6所示：

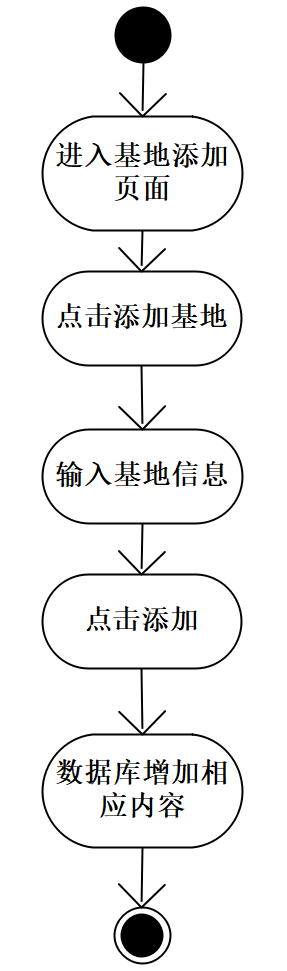


图2-6 研修室创建活动图

1. 用户前台研修室查询活动图如图2-7所示：

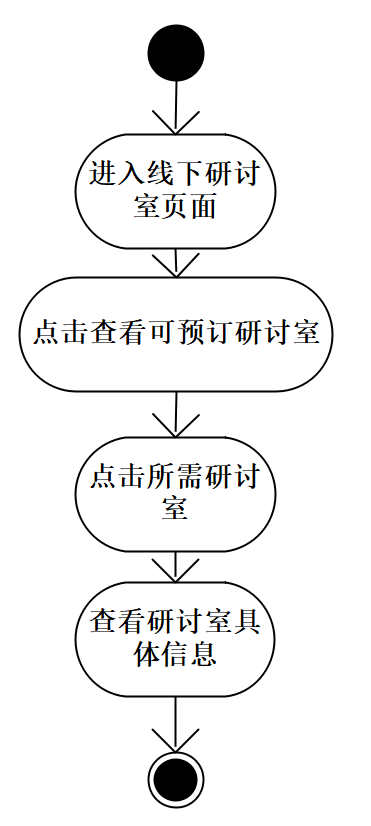


图2-7 研修室查询活动图

## 2.2 功能需求

### 2.2.1 角色分析

以下从协同创新总系统的实际需求分析，系统涉及到以下角色，如表2-1所示：

表2-1 系统角色表

|  |  |
| --- | --- |
| 角色 | 职责或功能 |
| 用户 | 个人公告的创建，进行问答，调查问卷的创建与填写，线下研修室的预定与退定，论坛帖子的创建与回帖 |
| 系统管理员 | 管理和维护整个系统的用户组织结构，负责对用户、角色、用户级别的增、删、改、查等管理。 |

### 2.2.2 业务功能

系统的总体用例图包含用户管理、信息发布、咨询反馈、线下研修室、网上调查、在线协同和技术论坛等用例，如下图2-8、2-9所示：

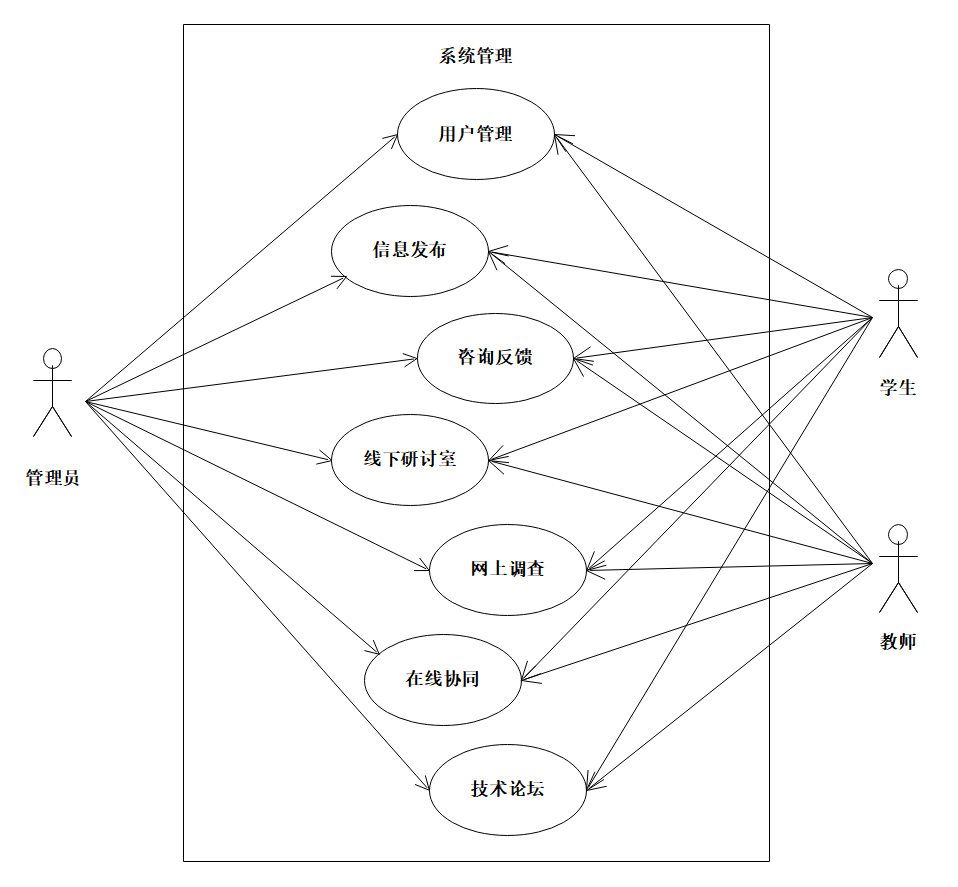


图2-8 系统总体用例图

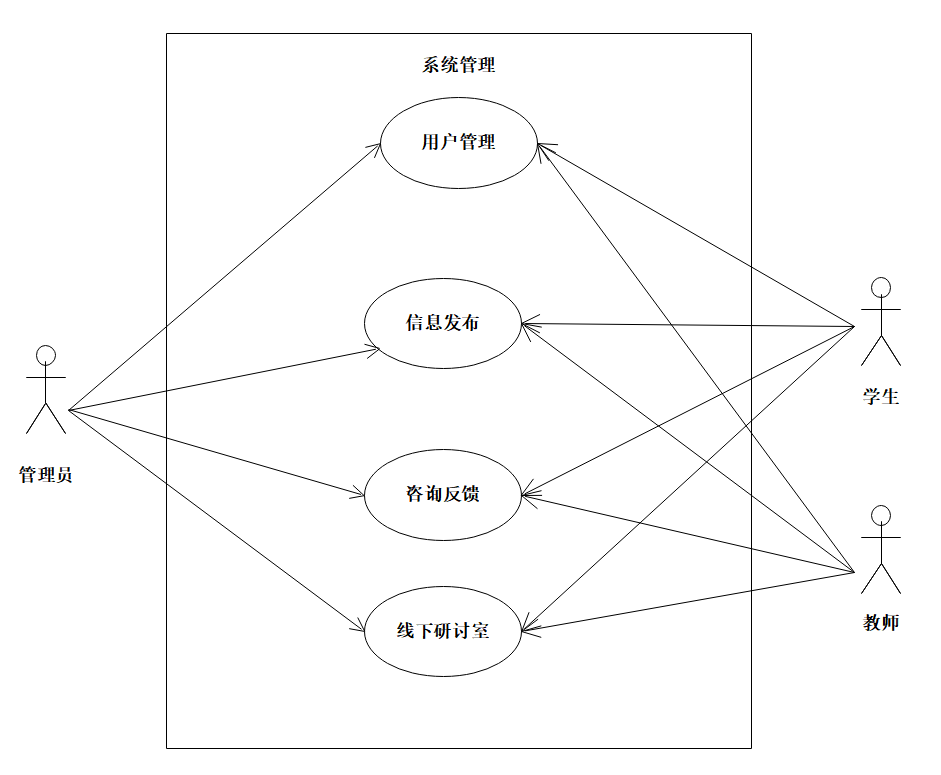


图2-9 高校协同创新信息门户子系统总用例图

2.2.2.1用户管理

用户管理用例图如下图2-10所示：

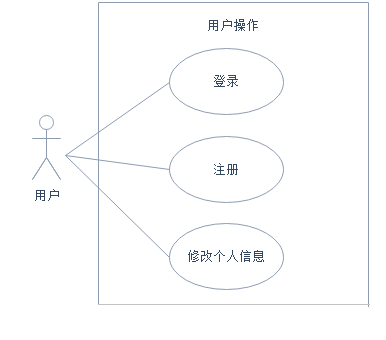


图2-10 用户管理用例图

用户进入个人信息页后左侧有资料查询、密码修改、个人信息修改、我的公告查询、我的公告删除、我的提问查询、我的提问删除、我的回答查询、我的回答删除、我的调查问卷查询、我的调查问卷删除等按钮，点击其中一个可查看到数据列表，每条数据都可点击右侧垃圾箱小按钮进行删除。

用户登录用例如下表2-2所示：

表2-2 用户登录用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0001 | 用例名称 | 登录 |
| 用例描述 | 登录是系统进行身份验证的功能模块，用户在提供有效的账号和密码后，发送系统后台进行身份验证，返回验证后的信息 | | |
| 执行者 | 学生，教师，管理员 | | |
| 前置条件 | 用户进入登录界面。 | | |
| 后置条件 | 用户进入平台首页。 | | |
| 涉众利益 | 1.使用者担心密码泄露  2.用户希望登录成功后能够记住密码 | | |
| 用例场景 | 系统登录界面 | | |
| 基本事件流 | 基本流程（Base Flow）   1. 填写账号、密码及验证码 2. 系统验证账号、密码及验证码是否填写完整 3. 将有效的用户账号、密码及验证码提交到服务器验证 4. 如果验证通过，直接登录到系统首页   扩展流程（Extend Flow）   1. 登录信息不充分： 2. 系统提示登录信息不充分 3. 用例结束 4. 账号或密码输入错误： 5. 提示账号或密码有误，请重新输入 6. 用例结束   字段列表（Filed List）  用户登录信息＝账号（邮箱或手机号）+密码+验证码  业务规则（Business role）   1. 账号不能为空，最长不能超过20个字符，为邮箱格式； 2. 密码不能为空，最长不能超过20个字符； 3. 密码必须是加密； | | |
| 特殊需求（Special Requirement） | 用户登录时系统响应时间应在5秒之内。 | | |

用户注册用例如下表2-3所示：

表2-3 用户注册用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0002 | 用例名称 | 注册 |
| 用例描述 | 注册是系统将信息存入数据库，用户在填写有效的账号和密码后，发送系统后台进行账号查重，返回查找后的信息，根据返回信息显示注册是否成功。 | | |
| 执行者 | 学生，教师 | | |
| 前置条件 | 用户进入注册界面。 | | |
| 后置条件 | 用户进入登录界面。 | | |
| 涉众利益 | 1.用户希望流程简洁明了。 | | |
| 用例场景 | 系统注册界面 | | |
| 基本事件流 | 基本流程（Base Flow）   1. 用户填写邮箱作为账号并设置密码，填写验证码后点击注册按钮 2. 系统验证账号、密码及验证码是否填写完整、有效 3. 若有效则要求用户选择身份为教师或学生 4. 根据选择结果跳转至不同详细资料填写界面 5. 填写完成并提交后系统将数据写入数据库 6. 注册成功，用户进入登录页面   扩展流程（Extend Flow）   1. 用户输入邮箱在数据库中已存在： 2. 系统提示该邮箱已被注册 3. 用例结束 4. 验证码错误： 5. 提示重新输入 6. 用例结束   字段列表（Filed List）  用户信息＝用户编号+用户名+密码+性别+年龄+手机号+邮箱  业务规则（Business role）   1. 账号不能为空，不能已被注册，最长不能超过20个字符，为邮箱格式； 2. 密码不能为空，最长不能超过20个字符； | | |
| 特殊需求（Special Requirement） | 用户注册时系统响应时间应在5秒之内。 | | |

用户修改个人信息用例如下表2-4所示：

表2-4 用户修改个人信息用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0003 | 用例名称 | 修改个人信息 |
| 用例描述 | 修改个人信息是用户修改个人信息并提交后，系统修改数据库该条数据并保存。 | | |
| 执行者 | 学生，教师 | | |
| 前置条件 | 用户进入个人资料界面，点击修改按钮。 | | |
| 后置条件 | 用户个人信息修改成功。 | | |
| 涉众利益 | 用户希望流程简洁明了。 | | |
| 用例场景 | 系统个人资料界面 | | |
| 基本事件流 | 基本流程（Base Flow）   1. 用户在首页中点击个人资料进入该页面 2. 用户在个人资料页面点击修改按钮 3. 用户修改个人资料并提交 4. 系统修改数据库该条数据并保存 5. 页面提示用户资料修改成功   扩展流程（Extend Flow）   1. 用户修改资料违规（如修改密码超出20字符）： 2. 系统提示修改失败 3. 用例结束   字段列表（Filed List）  用户信息＝用户编号+用户名+密码+性别+年龄+手机号+邮箱  业务规则（Business role）  密码不能为空，最长不能超过20个字符。 | | |
| 特殊需求（Special Requirement） | 用户修改资料时系统响应时间应在10秒之内。 | | |

用户查看通知消息用例如下表2-5所示：

表2-5 用户查看通知消息用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0004 | 用例名称 | 查看通知消息 |
| 用例描述 | 查看通知消息是系统进行查找和删除的功能模块，用户在个人主页可以查看通知消息、删除通知消息。 | | |
| 执行者 | 学生，教师 | | |
| 前置条件 | 用户进入个人资料界面。 | | |
| 后置条件 | 用户进入个人通知页面。 | | |
| 涉众利益 | 用户界面友好，简洁大方 | | |
| 用例场景 | 个人通知界面 | | |
| 基本事件流 | 基本流程（Base Flow）   1. 用户在首页中点击个人资料进入该页面 2. 用户在个人资料页面点击“我的消息”按钮显示个人通知页面   字段列表（Filed List）  用户个人通知＝标题+内容+时间+发布人  业务规则（Business role）  通知消息内容不能为空。 | | |
| 特殊需求（Special Requirement） | 用户查看通知时系统响应时间应在10秒之内。 | | |

2.2.2.2信息发布

管理员后台信息发布用例图如下图2-11所示：

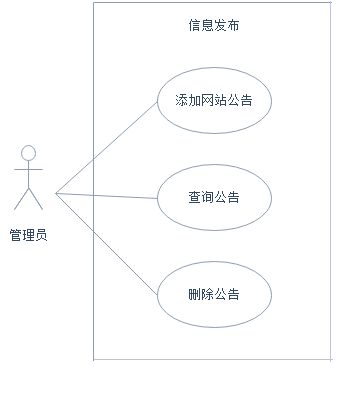


图2-11 管理员后台信息发布用例图

管理员进入信息发布页面，可在导航栏的下拉框选择个人公告与网站公告查看最新的公告，并且可以对违规公告进行删除，在搜索栏输入关键字进行查询，也可进行网站公告页面创建网站公告。

管理员网站公告添加用例如下表2-6所示：

表2-6 管理员网站公告添加用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0005 | 用例名称 | 网站公告添加 |
| 用例描述 | 网站公告添加是管理员操作的功能模块，管理员在成功登录后可以进入公告专栏点击添加公告。 | | |
| 执行者 | 管理员 | | |
| 前置条件 | 确认该登录用户为管理员 | | |
| 后置条件 | 操作成功，新增的公告在公告栏被显示。 | | |
| 涉众利益 | 1. 管理员担心数据出错或者文章乱码 2. 管理员希望文章不会乱序，能够按输入的格式显示在公告栏。 | | |
| 用例场景 | 公告栏页面 | | |

续表2-6

|  |  |
| --- | --- |
| 基本事件流 | 基本流程（Base Flow）   1. 点击网站公告 2. 点击添加公告按钮 3. 进入添加页面，填写公告内容 4. 操作提交至服务器   扩展流程（Extend Flow）   1. 公告无标题： 2. 系统提示公告缺少标题 3. 返回新增页面 4. 公告无内容： 5. 提示公告缺少内容 6. 返回新增页面   字段列表（Filed List）  公告显示信息＝标题+内容+时间+提交人  业务规则（Business role）   1. 公告内容字数不得少于30字，标题必须简练。 2. 内容描述必须清晰合理。 3. 不得多次提交 |
| 特殊需求（Special Requirement） | 公告可以是富文本格式 |

管理员后台公告查询用例如下表2-7所示：

表2-7 管理员后台公告查询用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0006 | 用例名称 | 公告查询 |
| 用例描述 | 公告查询模块是用户/管理员操作的功能模块，管理员在成功登录后可以进入信息发布页选择个人公告或网站公告，点击进入后将展示最新的公告。 | | |
| 执行者 | 管理员 | | |

续表2-7

|  |  |
| --- | --- |
| 前置条件 | 管理员登录成功 |
| 后置条件 | 数据库中公告信息展示在页面上 |
| 涉众利益 | 管理员担心数据加载过慢 |
| 用例场景 | 管理员进入公告页面 |
| 基本事件流 | 基本流程（Base Flow）   1. 进入首页 2. 点击进入公告页面 3. 页面向后台发送请求 4. 数据库中返回数据 5. 将数据刷新到页面   扩展流程（Extend Flow）   1. 公告分页： 2. 点击下一页将展示下一页的内容 3. 点击页号将展示对应页内容   字段列表（Filed List）  公告显示信息＝标题+内容+时间+提交人  业务规则（Business role）   1. 显示包含标题、时间以及部分内容 2. 按照时间顺序进行排序 |
| 特殊需求（Special Requirement） | 调到最后一页时下一页变灰 |

管理员后台公告删除用例如下表2-8所示：

表2-8 管理员后台公告删除用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0007 | 用例名称 | 公告删除 |
| 用例描述 | 公告删除模块是管理员操作的功能模块，管理员进入后台后查看公告，对于违规的公告，进行删除 | | |
| 执行者 | 管理员 | | |
| 前置条件 | 管理员进入公告页面，页面展示出所有公告，公告有违规内容 | | |
| 后置条件 | 公告被删除 | | |
| 涉众利益 | 管理员担心误删公告 | | |
| 用例场景 | 管理员对违规公告进行处理 | | |

续表2-8

|  |  |
| --- | --- |
| 基本事件流 | 基本流程（Base Flow）   1. 用户进入公告页面 2. 点击查看个人公告 3. 进入公告详情 4. 确认公告违规后进行删除 5. 数据库更新   扩展流程（Extend Flow）   1. 删除： 2. 弹出提示框是否删除 3. 点击否则不进行处理   字段列表（Filed List）  公告显示信息＝标题+内容+时间+提交人  业务规则（Business role）   1. 最新公告会显示在最上面 2. 按照时间顺序进行排序 |
| 特殊需求（Special Requirement） | 无 |

2.2.2.3咨询反馈

咨询反馈用例图如下图2-12所示：

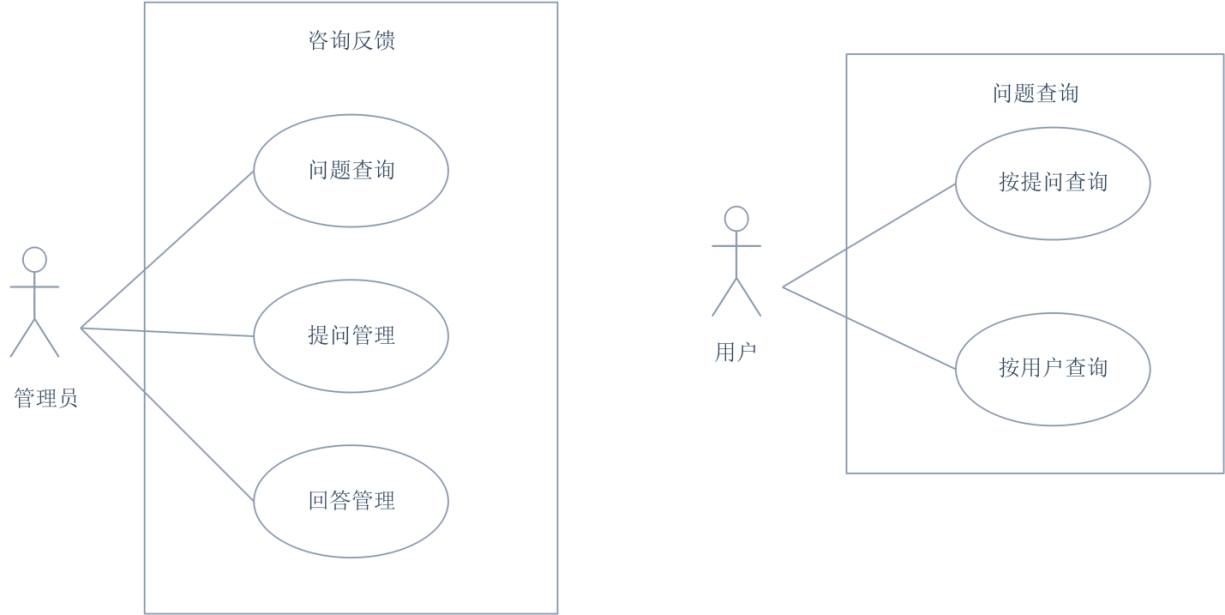


图2-12 咨询反馈用例图

用户进入咨询反馈板块，页面加载出最新提出的问题，用户可以通过关键字进行搜索，找到想要查看的问题后点击问题标题进入详细页面，可在详细页面进行回复，点赞，举报等操作。也可以点击搜索栏旁边的提问按钮，添加自己想要提出的问题。

问题查询用例描述如下表2-9所示：

表2-9 问题查询用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0008 | 用例名称 | 问题查询 |
| 用例描述 | 在咨询反馈页面的搜索框中输入搜索关键字，显示出问题中包含该关键字的提问。 | | |
| 执行者 | 用户 | | |
| 前置条件 | 进入咨询页面 | | |
| 后置条件 | 搜索结果显示在该页面 | | |
| 涉众利益 | 1.用户担心搜索无结果 | | |
| 用例场景 | 用户搜索问题 | | |
| 基本事件流 | 基本流程（Base Flow）   1. 用户进入咨询反馈页面 2. 输入关键字 3. 点击搜索 4. 查看问题   扩展流程（Extend Flow）   1. 若无结果： 2. 显示无结果 3. 结果过多： 4. 系统对结果进行分页 5. 每页显示10个   字段列表（Filed List）  提问 = 提问编号+提问者编号+提问时间+提问标题+提问内容  业务规则（Business role）   1. 最新提问会显示在最上面 2. 按照时间顺序进行排序 | | |
| 特殊需求（Special Requirement） | 能够防止注入攻击 | | |

提问用例描述如下表2-10所示：

表2-10 提问用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0009 | 用例名称 | 提问 |

续表2-10

|  |  |
| --- | --- |
| 用例描述 | 用户进入咨询反馈，点击提问，在输入页面输入问题，点击提交，提交成功后在提问界面显示该提问，用户进入个人信息页，点击我的提问，在我的提问中删除某一条。 |
| 执行者 | 用户 |
| 前置条件 | 用户登录成功 |
| 后置条件 | 提问提交或删除成功 |
| 涉众利益 | 用户担心提问没人回复  用户担心短时间内提问人数过多，自己的提问被刷下去 |
| 用例场景 | 用户在咨询反馈页面添加提问或在个人信息页删除提问 |
| 基本事件流 | 基本流程（Base Flow）   1. 用户进入咨询反馈页面 2. 点击添加提问 3. 填写提问内容 4. 点击提交 5. 数据库更新 6. 用户进入个人信息页面 7. 点击删除提问 8. 数据库更新   扩展流程（Extend Flow）   1. 若无人回答： 2. 该提问在一定时间内置顶 3. 建议提问人重发 4. 有人发布无关或违规回复： 5. 能够进行举报 6. 管理员接到通知后进行处理   字段列表（Filed List）  提问＝提问编号+提问者编号+提问时间+提问标题+提问内容  业务规则（Business role）   1. 最新提问会显示在最上面 2. 可对违规提问或回答进行举报 |
| 特殊需求（Special Requirement） | 按回答次数进行一定的排序优化 |

回答用例描述如下表2-11所示：

表2-11 回答用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0010 | 用例名称 | 回答问题 |
| 用例描述 | 用户进入咨询反馈页面，点击问题，进入问题详情，点击我要回答按钮，填写数据后进行提交。 | | |
| 执行者 | 用户 | | |
| 前置条件 | 用户进入问题详情页面 | | |
| 后置条件 | 回答被显示在问题页面 | | |
| 涉众利益 | 用户担心回答别人看不见 | | |
| 用例场景 | 用户进行问题回答 | | |
| 基本事件流 | 基本流程（Base Flow）   1. 用户进入咨询反馈页面 2. 点击问题，进入问题详情 3. 点击“我要回答”按钮 4. 填入数据 5. 点击提交   扩展流程（Extend Flow）   1. 若提问被删除： 2. 回复会不会被删除，但点击提问会显示提问已被删除 3. 点击删除时会提示是否确定 4. 回复内容违规或与问题无关： 5. 管理员删除回复 6. 通知回复人该回复已被删除   字段列表（Filed List）  回复＝问题编号+回答者编号+回答内容+回答时间  业务规则（Business role）   1. 回复内容应该符合提问内容 2. 对于非法回复管理员能够删除 3. 回复不能为空 | | |
| 特殊需求（Special Requirement） | 提交回复时会进行一次提问查找，防止回复提交时提问刚好被删除 | | |

2.2.2.4线下研修室

线下研修室用例图如下图2-13所示：

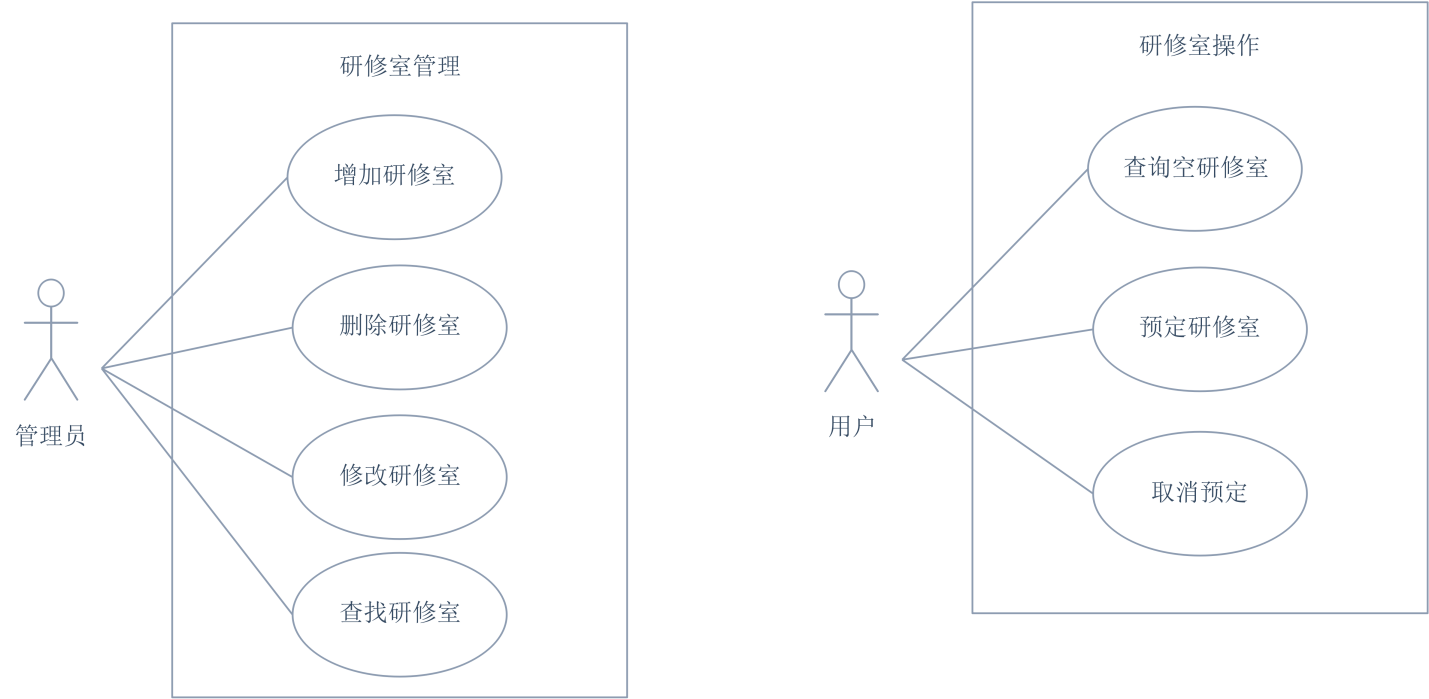


图2-13 线下研修室用例图

用户进入线下研修室页面，点击选中某个总研修室，进入后将展示所有可预定的子研修室，也可通过类型、人数、时间定位到符合条件的子研修室，点击预定后将弹出时间选择，选中想要预约的时间后点击确定，提交预定。管理员操作页面可点击添加研修室进行添加，也可点击修改对研修室数据进行修改，进入研修室详细页面，可点击添加进行子研修室添加，点击修改对子研修室数据进行修改。

线下研修室用例描述如下表2-12所示：

表2-12 线下研修室用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0011 | 用例名称 | 基地管理模块 |
| 用例描述 | 管理员进入基地管理板块，进行基地操作 | | |
| 执行者 | 管理员 | | |
| 前置条件 | 管理员进入线下研讨室板块 | | |
| 后置条件 | 数据库中新增/删除/修改基地信息 | | |
| 涉众利益 | 管理员担心有人同时添加该基地  管理员担心该基地已经被其他管理员添加 | | |
| 用例场景 | 用户删除个人回复 | | |

续表2-12

|  |  |
| --- | --- |
| 基本事件流 | 基本流程（Base Flow）   1. 管理员进入基地操作页面 2. 点击添加/删除/修改基地 3. 输入基地信息 4. 点击添加删除/修改 5. 操作成功   扩展流程（Extend Flow）   1. 若该基地重复： 2. 返回添加失败 3. 显示已存在的基地信息   字段列表（Filed List）  基地＝基地编号+基地地址+管理员id+添加时间  业务规则（Business role）   1. 基地地址真实存在 2. 非管理员无法操作 3. 地址不能为空 |
| 特殊需求（Special Requirement） | 在添加时与其他管理员进行沟通，保证不会出现重复添加 |

线下研修室操作用例描述如下表2-13所示：

表2-13 线下研修室操作用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0012 | 用例名称 | 研修室操作模块 |
| 用例描述 | 用户进入线下研修室板块 | | |
| 执行者 | 用户 | | |
| 前置条件 | 用户进入线下研修室板块 | | |
| 后置条件 | 显示所有研修室 | | |
| 涉众利益 | 用户担心没有研修室可用  管理员担心有人预定却不来，浪费资源 | | |
| 用例场景 | 用户查看研修室 | | |

续表2-13

|  |  |
| --- | --- |
| 基本事件流 | 基本流程（Base Flow）   1. 1. 用户进入线下研修室页面 2. 2. 点击查看可预订研修讨室 3. 3. 点击某一研修室 4. 4. 显示具体信息   扩展流程（Extend Flow）   1. 若该基地重复： 2. 返回添加失败 3. 显示已存在的研修室信息   字段列表（Filed List）  研修室＝房间编号+从属研修室+户型+申请人编号+占用时间段  业务规则（Business role）   1. 不可一人多定 2. 预定必须到，若有情况，提前取消 3. 占用时间段限制 |
| 特殊需求（Special Requirement） | 不得申请与实际人数相差过大的研修室 |

## 2.3 非功能需求

### 2.3.1环境需求

服务器端运行环境需求如下表2-14所示：

表 2-14 服务器端环境

|  |  |
| --- | --- |
| **服务器端运行环境** | |
| **软件** | **版本要求** |
| 浏览器 | 火狐、IE 10以上 |
| JDK | 1.8 |
| Apache Tomcat | 9.0.14 |
| IDEA | 2018.3 |
| Mysql | 8.0 |

客户端运行环境需求如下表2-15所示：

表 2-15 客户端环境

|  |  |
| --- | --- |
| **客户端运行环境** | |
| **软件** | **版本要求** |
| 浏览器 | 火狐、IE 10以上 |
| 操作系统 | Win7及以上 |

### 2.3.2 性能需求

1. 可维护性
2. 维护方法：系统能在数据存储出错时进行自动回滚，通过正则表达式对输入数据进行判定并进行处理。
3. 易操作性
4. 界面设计：系统应提供美观、实用、方便和直观的图形用户管理界面，保证不论是年轻大学生还是年老教师都能够在极短时间内了解并学会使用网站功能，所有功能简明易懂，不会有太繁琐的操作。
5. 可扩展性
6. 功能扩展：本系统从系统的总体结构入手，按照用户与管理员的需求进行扩展，保留部分接口，以满足开未来用户对其他功能的需要。
7. 软硬件升级：本系统由于采用B/S架构，所以对用户的硬件需求几乎没有，系统则需要一定的配置，在硬件上若不满足则需要将项目部署到更好的服务器上，保证数据库等软件保持一个较高的版本。
8. 开放性
9. 该系统在进行多数据传输时，将数据打包成JSON字符串，保证了在任何计算机上都能正常识别。

### 2.3.3 安全需求

1. 容错性

输入框能够验证用户输入内容，通过正则表达式对用户输入进行验证，系统拥有一定的错误处理功能。在出错的情况下，系统能在一定程度解决错误并提醒用户，提示用户重新输入或者进行自动的修复校正。

1. 系统的外在环境安全

系统要保证能有一定的访问抗压性，能够承受一定量的访问压力，数据库要每隔一定时间就做好备份；如果出现了软件或者硬件故障等紧急情况，能够自动的跳转到备用服务器上，保证用户的正常使用。

1. 系统内部安全

在不出现外部安全情况下，该系统还要保证用户的安全使用，保证不会出现无权限操作等情况的发生。

1. 系统运行安全

该系统应具有抵抗非法入侵的能力，对非法访问进行拦截；在物理方面，该系统应确保没有潜在的单点故障，并提供资源的数据备份功能。保证能够抵御注入攻击，一定程度上防御dos攻击，系统支持定期自动和手动数据备份，操作出错能够进行回滚，能够在数据损坏或数据丢失的情况下找回数据，实现一定程度的数据恢复。

3 总体设计

## 3.1系统设计的原则

在充分认识了该系统的功能需求，以及该系统的使用情景，本系统应该严格按照以下几点进行设计：

1. 实用性原则。该系统面向广大学生与教师，旨于解决学生的问题与教师的需求。本系统力求能够快速即时地传递大学生之间的问题与需求，彻底解决传统协同合作方式的效率低下问题。
2. 先进性原则。在强调实用性的同时，我们还要考虑系统的先进性。在即将来临的5G时代，信息化发展的速度越来越快，对此我们不能抱着当下的技术一成不变，必须紧跟时代，不断进步，不断完善项目，更新技术，保证系统时刻充满着新鲜的血液。所以，为了与时俱进，有一个较高的起点，在本系统的设计过程中我采用目前较为先进和成熟的技术和理念，使系统在技术上拥有一定的优势。
3. 模块化原则。模块作为操作的集合、功能的集合、对象的集合，清晰的边界能够使项目的功能扩展更加轻松，模块的作用更加明确，大大降低了后期维护的成本。模块化将一个待实现的大功能分解成若干个小的简单操作，每个模块可独立开发、测试，最后组装成完整的程序，承载着分而治之的思想。模块化使程序的结构更加清晰，便于程序员的阅读、理解、测试和修改。
4. 安全性原则。系统提供安全机制以防止非法授权操作，为各子系统提供严格灵活的用户和权限划分，对于数据库管理，建立起完备的数据备份恢复机制。对关键的信息的操作提供日志记录。
5. 可扩展性原则。出于对后续系统升级的考虑，系统的所有功能模块的设计都预留了扩展接口。因为网站的需求和应用目标会随着技术进步和未来形式的变化而发生转变，在设计初期就考虑到可能发生的变化，是十分必要的。

## 3.2 系统体系结构设计

浏览器-服务器( Browser/Server， B/S)体系结构如下图3-1所示。

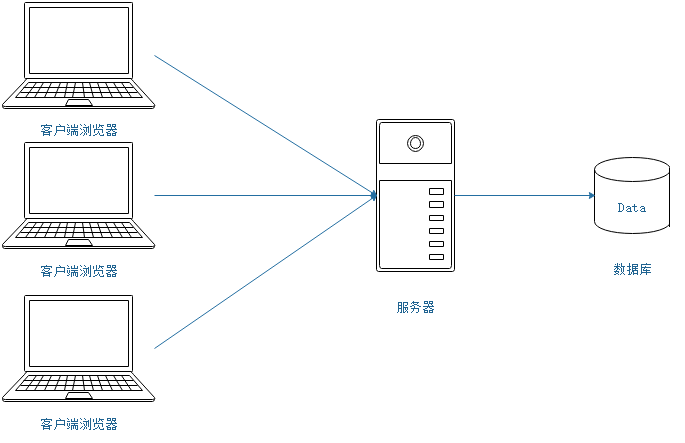


图3-1 B/S体系结构

1. B/S体系结构相对于C/S结构所占空间更少，使用更方便，不需下载软件以及后续不断地更新。
2. 在B/S体系结构中，应用程序所有文件都是存放于Web服务器上，降低了用户本地的硬件负担，用户浏览网页，操作功能，只是向浏览器发送请求，正真的操作在后台服务器，用户发出请求后服务器进行处理，最后把结果展示在页面上，数据库进行操作完成相应的数据处理工作，最后将结果通过浏览器展示给用户。
3. 以B/S架构部署的项目，更新、配置和维护全在服务器端解决。用户在使用系统时，不会有频繁的更新，项目不用花大量的精力去兼容各种运行环境，不用去应对各种错误，一个浏览器就能够满足用户的所有需求。在用户浏览网页时，项目的热更新，用户是察觉不到的，极大的提升了用户体验。B/S体系结构不限制移动端或PC端，不限制操作系统，也不限制浏览器的种类，适用面特别广泛。

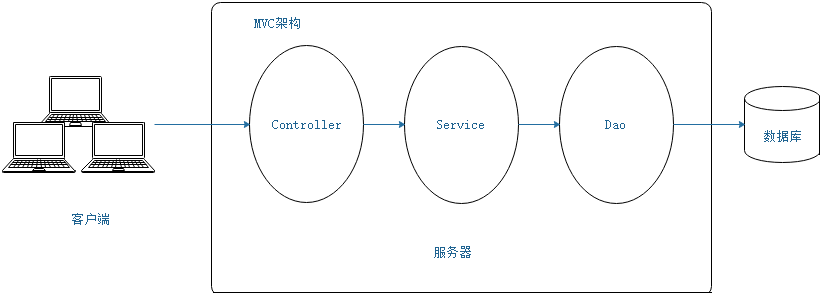


图3-2 系统结构图

如上图3-2所示，系统一共分为三层：视图层、控制层、业务逻辑层、数据层。其中视图层使用控制层、控制层使用业务逻辑层、业务逻辑层使用数据层。视图层主要包括展现界面的UI展示类，控制层负责功能调用，业务逻辑层实现一定的业务处理，数据层负责调用MyBatis中的对象，进行sql语句调用。视图层依赖Layui得以展现，数据层依赖MyBatis技术。三层结构基于MVC架构。

## 3.3 系统功能结构设计

系统功能架构设计定义软件系统各主要部件及成分之间的关系如下图3-3所示：

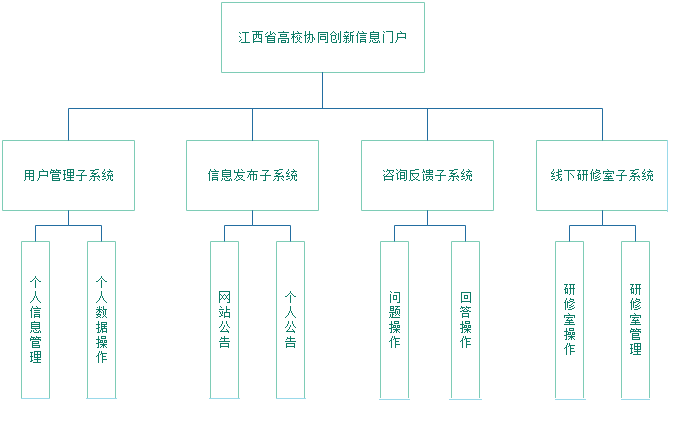


图3-3 系统功能结构图

协同创新平台共有 用户管理子系统、信息发布子系统、咨询反馈子系统、网上调查调查子系统、线下研修室子系统、在线协同子系统、技术论坛子系统 七个子系统。

我负责用户管理子系统、信息发布子系统、咨询反馈子系统、网上调查调查子系统，用户管理子系统主要包含个人信息的处理以及涉及到的各种信息的处理；信息发布子系统允许用户在该板块下查找想要的信息，提问以及回答，对违规信息举报，对优质回答进行点赞；网上调查子系统能收集用户的意见，推动网站的发展，个人也能进行调查问卷的创建，来获取自己想要的数据。

# **4 数据库设计**

## 4.1 概念结构设计

### 4.1.1 设计思路

在进行完需求分析之后，就要考虑其中实体所应承载的各种属性，设计阶段考虑的越周全，后期改动越少，越轻松，所以这一步非常重要。

首先，思考需要哪些实体来承载需要的数据，将所需数据进行统计，根据得出的数据实体进行建表，对于该项目，各个子系统下对应着一个或者多个实体，实体之间有的相互联系，有的相互独立，把握好它们之间的联系就能够初步定下实体所应有的隐藏属性、外键等。考虑到效率问题，适度加入索引，加快查询速度。

在数据库中创建的调查问卷表通过外键，将调查问题表挂在调查问卷表下，又将选项表挂在调查问题表下，最后形成了一个高度为3的N叉树；对于研讨室，在总的研讨室下挂着无数个子研讨室，而每个子研讨室都有无数的使用记录与三张使用状态表，使用记录又与用户id相关联，与使用状态表相关联。

### 4.1.2 E-R图

该系统的实体关系图E-R图如下图4-1所示：

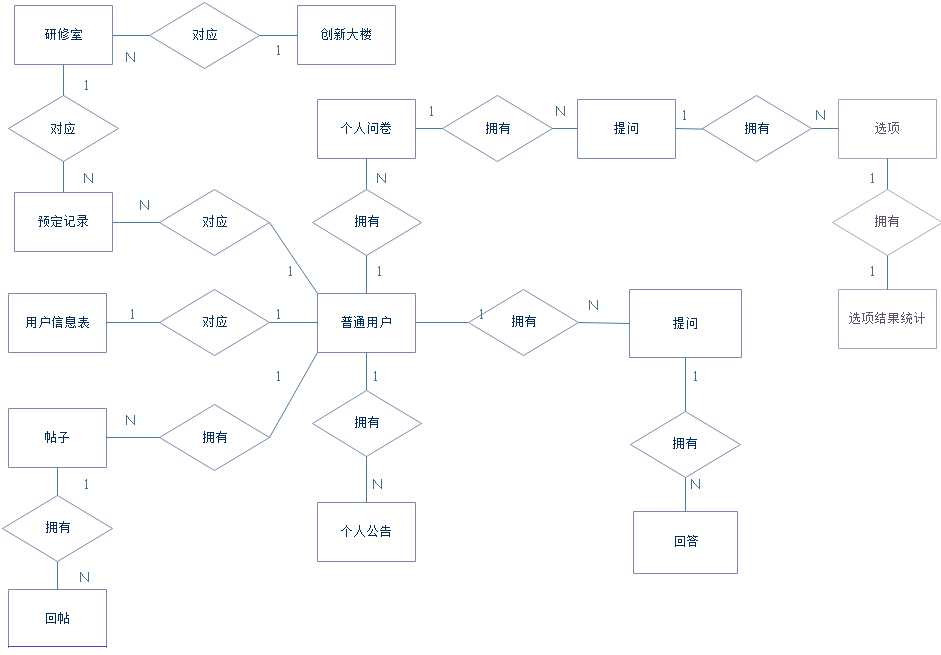


图4-1 实体关系图

### 4.1.3实体属性设计

1. 用户信息实体图，如图4-2所示：

实体：用户

属性：编号、用户名、身份编号、邮箱、手机号、年龄、性别、密码

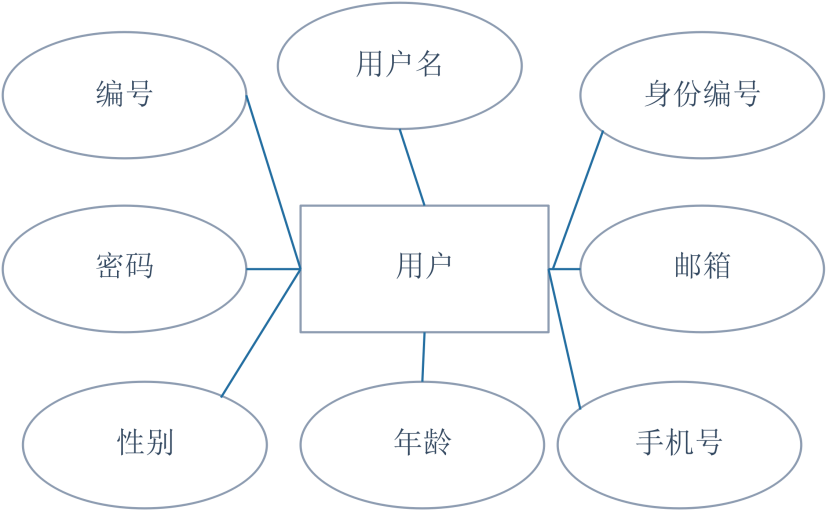


图4-2 用户信息实体图

1. 公告实体图，如图4-3所示：

实体：公告

属性：编号、标题、内容、时间、发布人

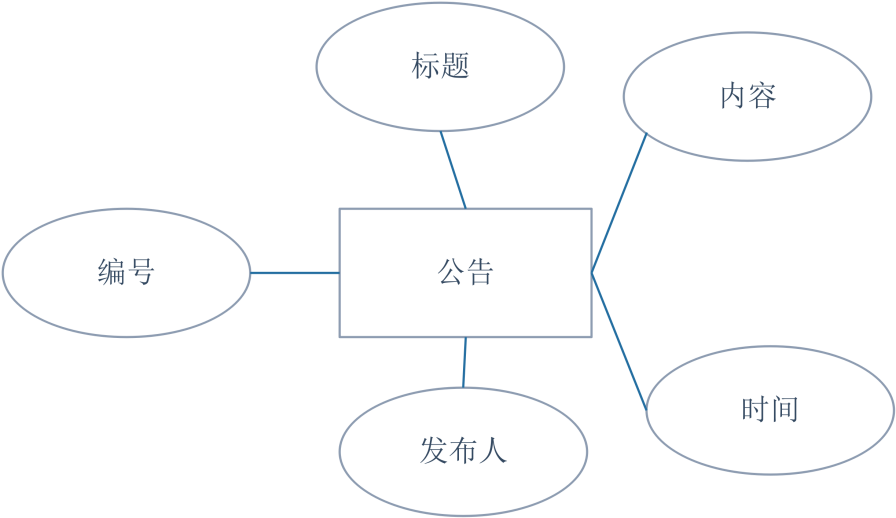


图4-3 公告实体图

1. 问题实体图，如图4-4所示：

实体：问题

属性：编号、提问者编号、提问时间、提问标题、提问内容

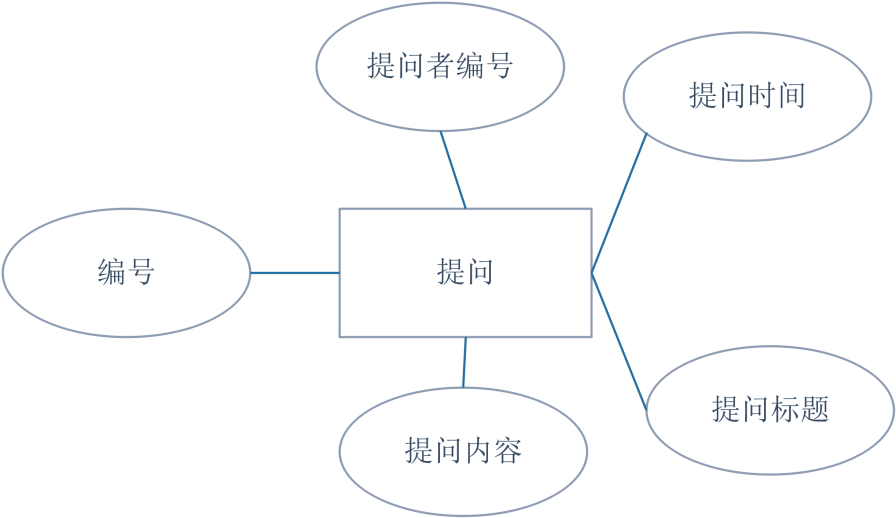


图4-4 问题实体图

1. 回答实体图，如图4-5所示：

实体：回答

属性：编号、回答者编号、问题编号、回答时间、回答内容

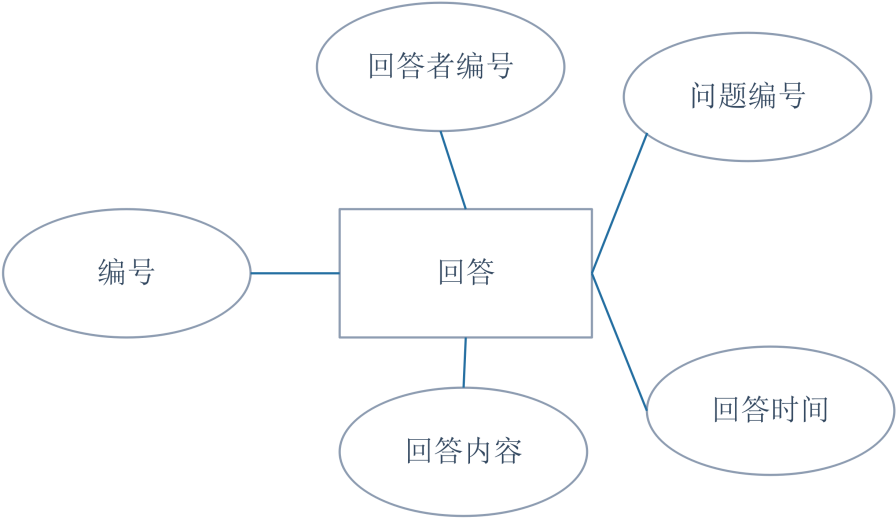


图 4-5 回答实体图

1. 研修室实体图，如图4-6所示：

实体：研修室

属性：编号、从属研修室id、类型、设备、名称、容纳人数、备注

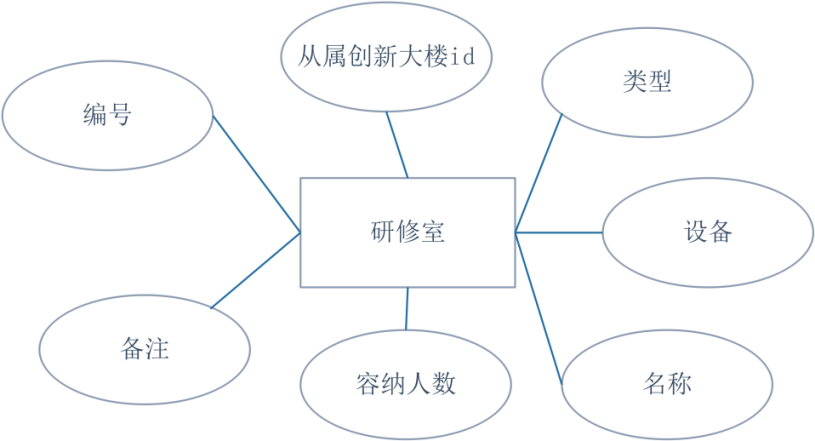


图4-6 研修室实体图

1. 研修室使用状态实体图，如图4-7所示：

实体：研修室使用状态

属性：编号、研修室编号、时间段、使用状态

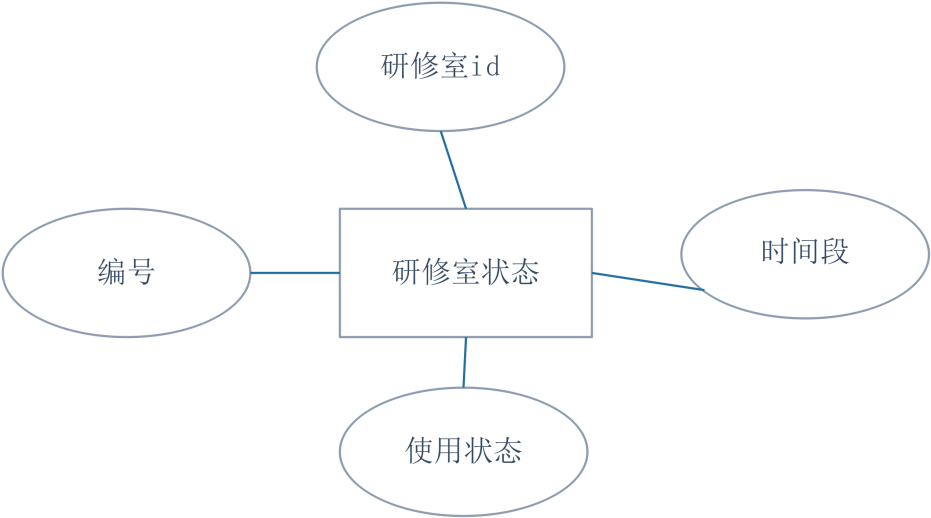


图4-7 研修室使用状态实体图

1. 预定记录实体图，如图4-8所示：

实体：预定记录

属性：编号、研修室编号、申请人编号、占用时间段、预定时间

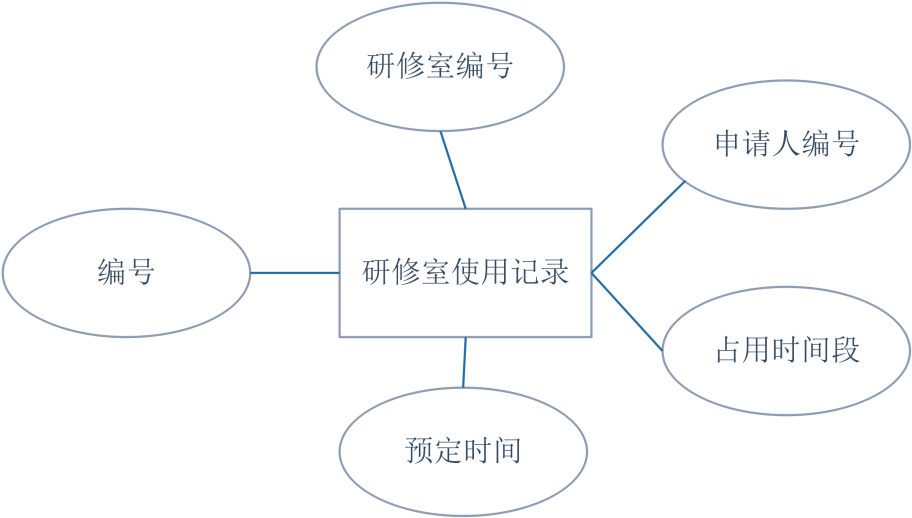


图4-8 预定记录实体图

## 4.2 逻辑结构设计

### 4.2.1 设计思路

（1）将相关联的表以图、树等结构进行设计

（2）根据数据可能需要的长度大小、内容要求来定义其使用类型

（3）按照第三范式将过大的表拆成较小的表

### 4.2.2 逻辑模型

1. 系统用户表,如下表4-9所示：

表4-9用户信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | id | 整型 | 11 | 是 | 用户编号 |
| 2 | name | 字符型 | 20 | 否 | 用户名称 |
| 3 | sex | 字符型 | 1 | 否 | 性别 |
| 4 | password | 字符型 | 20 | 否 | 登录密码 |
| 5 | tel | 字符型 | 11 | 否 | 电话号码 |
| 6 | age | 整型 | 2 | 否 | 年龄 |
| 7 | email | 字符型 | 20 | 否 | 邮箱 |
| 8 | detail | 字符型 | 11 | 否 | 详细资料字符串 |

1. 公告表，如下表4-10所示：

表4-10公告表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | id | 整型 | 11 | 是 | 公告编号 |
| 2 | title | 字符型 | 50 | 否 | 公告标题 |
| 3 | content | 文本型 | 65535 | 否 | 内容 |
| 4 | uid | 整型 | 11 | 否 | 发布人编号 |
| 5 | time | 时间戳 | 20 | 否 | 发布时间 |

1. 调查问卷表,如下表4-11所示：

表4-11调查问卷表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | did | 整型 | 11 | 是 | 实训班级编号 |
| 2 | title | 字符型 | 20 | 否 | 实训班级名称 |
| 3 | addTime | 时间戳 | 20 | 否 | 添加时间 |
| 4 | endTime | 时间戳 | 20 | 否 | 结束时间 |
| 5 | uid | 整型 | 11 | 否 | 用户编号，外键 |

1. 调查问卷问题表,如下表4-12所示：

表4-12调查问卷问题表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | dqid | 整型 | 11 | 是 | 问题编号 |
| 2 | did | 整型 | 11 | 否 | 调查问卷编号，外键 |
| 3 | title | 字符型 | 60 | 否 | 标题 |

1. 选项表,如下表4-13所示：

表4-13选项表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | oid | 整型 | 11 | 是 | 选项编号 |
| 2 | dqid | 整型 | 20 | 否 | 问题编号。外键 |
| 3 | content | 字符型 | 60 | 否 | 选项内容 |

1. 问卷提交表,如下表4-14所示：

表4-14问卷提交表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | dsid | 整型 | 11 | 是 | 提交编号 |
| 2 | did | 整型 | 10 | 否 | 问卷编号，外键 |
| 3 | uid | 整型 | 10 | 否 | 用户编号，外键 |
| 4 | submitTime | 时间戳 | 20 | 否 | 提交时间 |

1. 选项统计表,如下表4-15所示：

表4-15选项统计表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | sid | 整型 | 11 | 是 | 统计编号 |
| 2 | oid | 整型 | 20 | 否 | 选项编号，外键 |
| 3 | statisticalResult | 整型 | 10 | 否 | 统计结果 |

1. 提问表,如下表4-16所示：

表4-16提问表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | aid | 整型 | 11 | 是 | 提问编号 |
| 2 | title | 字符型 | 30 | 否 | 提问标题 |
| 3 | content | 字符型 | 10 | 否 | 提问内容 |
| 4 | uid | 整型 | 11 | 否 | 用户编号，外键 |
| 5 | time | 时间戳 | 20 | 否 | 提交时间 |

1. 回答表,如下表4-17所示：

表4-17回答表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | rid | 整型 | 11 | 是 | 回答编号 |
| 2 | aid | 整型 | 30 | 否 | 问题编号，外键 |
| 3 | uid | 整型 | 10 | 否 | 用户编号，外键 |
| 4 | content | 字符型 | 200 | 否 | 回答内容 |
| 5 | time | 时间戳 | 20 | 否 | 回答时间 |

1. 创新大楼表，如下表4-18所示：

表4-18创新大楼表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | hid | 整型 | 11 | 是 | 大楼编号 |
| 2 | school | 字符型 | 30 | 否 | 所属学校名称 |
| 3 | uid | 整型 | 10 | 否 | 用户编号，外键 |
| 4 | addTime | 时间戳 | 20 | 否 | 添加时间 |
| 5 | houseName | 字符型 | 45 | 否 | 大楼名称 |
| 6 | tips | 字符型 | 45 | 否 | 备注 |

1. 研修室表,如下表4-19所示：

表4-19研修室表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | rid | 整型 | 11 | 是 | 研修室编号 |
| 2 | hid | 整型 | 20 | 否 | 大楼编号，外键 |
| 3 | major | 字符型 | 11 | 否 | 作用类型 |
| 4 | roomName | 字符型 | 11 | 否 | 研修室名称 |
| 5 | num | 字符型 | 11 | 否 | 容纳人数 |
| 6 | equipment | 字符型 | 11 | 否 | 设备 |
| 7 | tips | 字符型 | 200 | 否 | 备注 |

1. 房间状态表,如下表4-20所示：

表4-20：房间状态表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | ruid | 整型 | 11 | 是 | 编号 |
| 2 | rid | 整型 | 10 | 否 | 研修室编号，外键 |
| 3 | time | 字符型 | 50 | 否 | 预约时间 |
| 4 | state | 整型 | 11 | 否 | 使用状态 |

1. 预定记录表,如下表4-21所示：

表4-21：预定记录表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | rurid | 整型 | 11 | 是 | 记录编号 |
| 2 | rid | 整型 | 20 | 否 | 研修室编号，外键 |
| 3 | uid | 整型 | 20 | 否 | 用户编号，外键 |
| 4 | time | 字符型 | 2 | 否 | 预约时间段 |
| 5 | addTime | 时间戳 | 20 | 否 | 添加时间 |

## 4.3 物理结构设计

### 4.3.1 存取方式

在数据库中，添加了定时任务，代码如下：

CREATE DEFINER=`root`@`localhost` PROCEDURE `fresh\_morning`()

begin

update roomuse set state = 1 where time = '上午';

end

CREATE DEFINER=`root`@`localhost` PROCEDURE `fresh\_afternoon`()

begin

update roomuse set state = 1 where time = '下午';

end

CREATE DEFINER=`root`@`localhost` PROCEDURE `fresh\_roomuse`()

BEGIN

SET SQL\_SAFE\_UPDATES = 0;

update roomuse set state = 0;

SET SQL\_SAFE\_UPDATES = 1;

END

CREATE EVENT `user\_event`

ON SCHEDULE

EVERY 1 DAY STARTS DATE\_ADD(DATE\_ADD(CURDATE(), INTERVAL 1 DAY), INTERVAL 7 HOUR) ON COMPLETION NOT PRESERVEENABLE

DO call fresh\_morning()

### 4.3.2 存储结构

1. 用户信息表，如下表4-22所示：

表4-22 用户信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | id | int | 11 | 是 | 用户编号 |
| 2 | name | varchar | 20 | 否 | 用户名称 |
| 3 | sex | varchar | 1 | 否 | 性别 |
| 4 | password | varchar | 20 | 否 | 登录密码 |
| 5 | tel | varchar | 11 | 否 | 电话号码 |
| 6 | age | int | 2 | 否 | 年龄 |
| 7 | email | varchar | 20 | 否 | 邮箱 |
| 8 | detail | varchar | 11 | 否 | 详细资料字符串 |

1. 公告表，如下表4-23所示：

表4-23 公告表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | id | int | 11 | 是 | 公告编号 |
| 2 | title | varchar | 50 | 否 | 公告标题 |
| 3 | content | text | 65535 | 否 | 内容 |
| 4 | uid | int | 11 | 否 | 发布人编号 |
| 5 | time | timestamp | 20 | 否 | 发布时间 |

1. 调查问卷表，如下表4-24所示：

表4-24 调查问卷表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | did | int | 11 | 是 | 实训班级编号 |
| 2 | title | varchar | 20 | 否 | 实训班级名称 |
| 3 | addTime | timeStamp | 20 | 否 | 添加时间 |
| 4 | endTime | timeStamp | 20 | 否 | 结束时间 |
| 5 | uid | int | 11 | 否 | 用户编号，外键 |

1. 调查问卷问题表，如下表4-25所示：

表4-25 调查问卷问题表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | dqid | int | 11 | 是 | 问题编号 |
| 2 | did | int | 11 | 否 | 调查问卷编号，外键 |
| 3 | title | varchar | 60 | 否 | 标题 |

1. 调查问卷选项表，如下表4-26所示：

表4-26 选项表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | oid | int | 11 | 是 | 选项编号 |
| 2 | dqid | int | 20 | 否 | 问题编号。外键 |
| 3 | content | varchar | 60 | 否 | 选项内容 |

1. 问卷提交表，如下表4-27所示：

表4-27问卷提交表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | dsid | int | 11 | 是 | 提交编号 |
| 2 | did | int | 10 | 否 | 问卷编号，外键 |
| 3 | uid | Int | 10 | 否 | 用户编号，外键 |
| 4 | submitTime | timeStamp | 20 | 否 | 提交时间 |

1. 选项统计表，如下表4-28所示：

表4-28选项统计表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | sid | int | 11 | 是 | 统计编号 |
| 2 | oid | int | 20 | 否 | 选项编号，外键 |
| 3 | statisticalResult | int | 10 | 否 | 统计结果 |

1. 提问表，如下表4-29所示：

表4-29 提问表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | aid | int | 11 | 是 | 提问编号 |
| 2 | title | varchar | 30 | 否 | 提问标题 |
| 3 | content | varchar | 10 | 否 | 提问内容 |
| 4 | uid | int | 11 | 否 | 用户编号，外键 |
| 5 | time | timestamp | 20 | 否 | 提交时间 |

1. 回答表，如下表4-30所示：

表4-30 回答表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | rid | int | 11 | 是 | 回答编号 |
| 2 | aid | int | 30 | 否 | 问题编号，外键 |
| 3 | uid | Int | 10 | 否 | 用户编号，外键 |
| 4 | content | varchar | 200 | 否 | 回答内容 |
| 5 | time | timestamp | 20 | 否 | 回答时间 |

1. 创新大楼表，如下表4-31所示：

表4-31创新大楼表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | hid | int | 11 | 是 | 大楼编号 |
| 2 | school | varchar | 30 | 否 | 所属学校名称 |
| 3 | uid | int | 10 | 否 | 用户编号，外键 |
| 4 | addTime | timestamp | 20 | 否 | 添加时间 |
| 5 | houseName | varchar | 45 | 否 | 大楼名称 |
| 6 | tips | varchar | 45 | 否 | 备注 |

1. 研修室表，如下表4-32所示：

表4-32研修室表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | rid | int | 11 | 是 | 研修室编号 |
| 2 | hid | int | 20 | 否 | 大楼编号，外键 |
| 3 | major | varchar | 11 | 否 | 作用类型 |
| 4 | roomName | varchar | 11 | 否 | 研修室名称 |
| 5 | num | varchar | 11 | 否 | 容纳人数 |
| 6 | equipment | varchar | 11 | 否 | 设备 |
| 7 | tips | varchar | 200 | 否 | 备注 |

1. 研修室状态表，如下表4-33所示：

表4-33房间状态表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | ruid | int | 11 | 是 | 编号 |
| 2 | rid | int | 10 | 否 | 研修室编号，外键 |
| 3 | time | varchar | 50 | 否 | 预约时间 |
| 4 | state | int | 11 | 否 | 使用状态 |

1. 研修室预定记录表，如下表4-34所示：

表4-34预定记录表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | rurid | int | 11 | 是 | 记录编号 |
| 2 | rid | int | 20 | 否 | 研修室编号，外键 |
| 3 | uid | int | 20 | 否 | 用户编号，外键 |
| 4 | time | varchar | 2 | 否 | 预约时间段 |
| 5 | addTime | timestamp | 20 | 否 | 添加时间 |

# 5 界面设计

## 5.1 界面关系图或工作流图

管理员进入网站后台后，可以对所有的公告、问题与回答、调查问卷、论坛帖子进行查看与删除，对研修室进行添加、删除、修改操作，对于举报进行处理。

管理员工作流图如下图5-1所示：

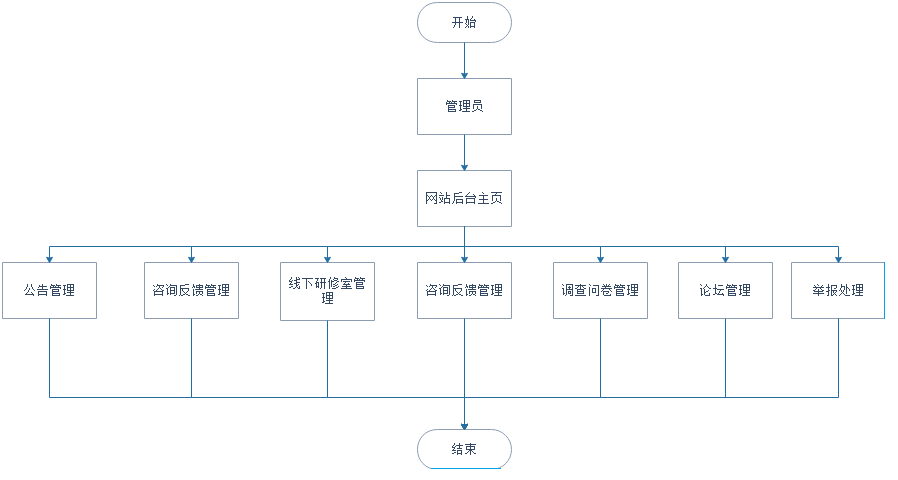


图5-1 管理员工作流图

用户进入网站首页后，可以创建个人公告、查看所有公告、提出问题、查看问题、对问题做出回答，创建调查问卷、填写调查问卷、查看研修室并进行预定退定，在论坛发帖、回帖。

用户工作流图如下图5-2所示：

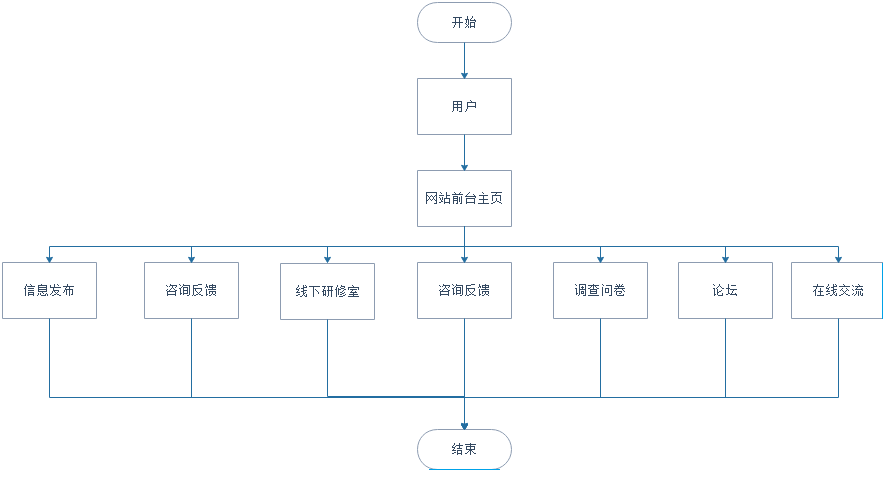


图5-2 用户工作流图

## 5.2 界面设计成果

### 5.2.1 主界面

后台主界面如下图5-3所示：



图5-3 后台主界面

前台主界面如下图5-4所示：



图5-4 前台主界面

### 5.2.2 子界面

（1）用户登录子界面

用户输入网址进入登录页面如下图5-5所示：



图5-5 前台登录界面

（2）用户咨询反馈子界面

用户登录成功后，点击导航栏的咨询反馈进入问题页面如下图5-6所示：



图5-6 前台问题界面

（3）用户线下研修室子界面

用户登录成功后，点击导航栏的线下研修室进入问题页面如下图5-7所示：



图5-7前台研修室页面

6 详细设计

## 6.1 系统主要功能模块介绍

系统主要功能模块如下图6-1所示：

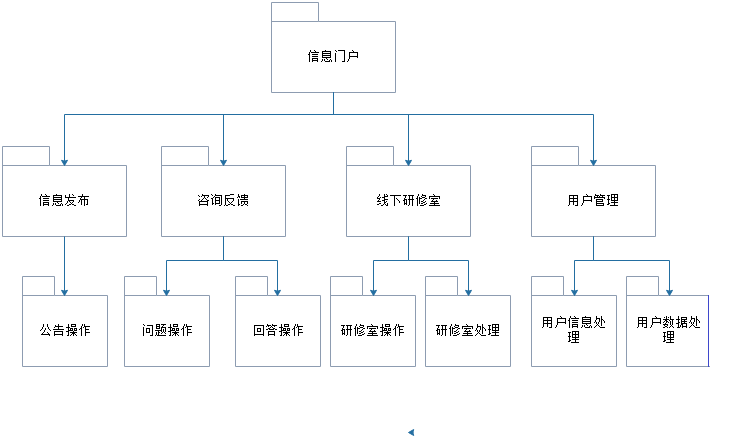


图6-1 系统总包图

1. 用户子系统：
2. 用户信息操作：用户拥有信息查询，信息修改，密码修改功能。
3. 用户数据处理：对用户添加的问答、调查问卷、公告等进行查询、删除。
4. 信息发布子系统：

公告操作：查询公告、创建公告，管理员删除公告、管理员创建公告。

1. 咨询反馈子系统：
2. 问题操作：查询问题、关键字查询、举报、管理员删除问题
3. 回答操作：回复问题，回答点赞、回答举报、管理员删除回答
4. 线下研讨室子系统：
5. 研讨室操作：管理员添加、修改、删除研修室，用户查找、预定、退定研讨室。
6. 子研讨室操作操作：用户预定、退定、管理员添加、删除、修改。

## 6.2功能模块设计

### 6.2.1用户管理

6.2.1.1 功能结构设计

用户管理包图如下图6-2所示：

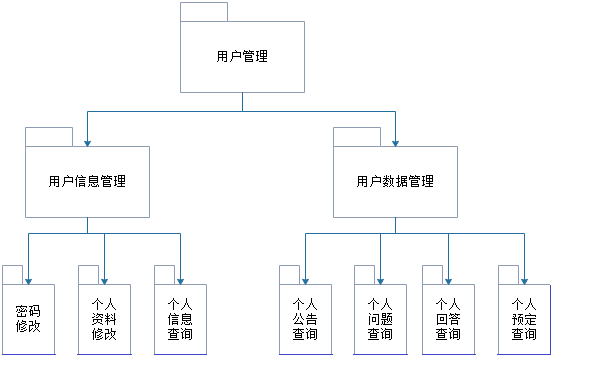


图6-2 用户管理包图

用户管理子系统包括“用户信息管理”、“用户数据管理”两个模块，“用户信息管理”模块主要功能是对用户的个人信息进行查询与修改，用户可以在个人中心页面看到自己注册时的信息，并且对其中的部分数据进行修改，也可以对密码进行修改。“用户数据管理”模块则是处理用户在网站中交互产生的数据，如：提问、回答、调查问卷填写等，可以对预定的研修室取消预约。

6.2.1.2 类图设计

用户控制类图如下图6-3所示：

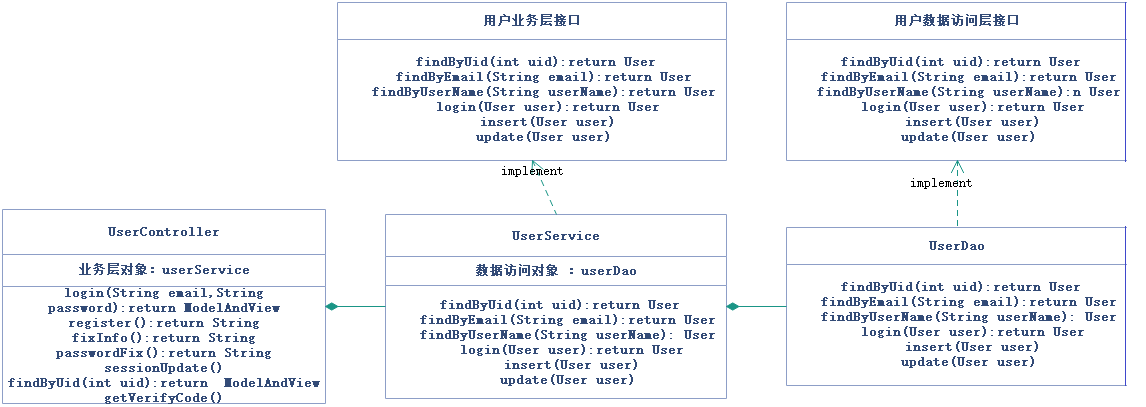


图6-3 用户控制类图

用户控制类描述如下表6-1所示：

表6-1 用户控制类描述

|  |  |  |
| --- | --- | --- |
| 用户控制类 | | |
| 类的描述：负责用户的登录、注册验证，获取个人信息、验证码创建 | | |
| 操作描述 | login () | 登录验证 |
| register() | 注册 |
| fixInfo () | 修改个人信息 |
| passwordFix() | 密码修改 |
| getVerifyCode() | 验证码获取 |

用户服务类描述如下表6-2所示：

表6-2 用户服务类描述

|  |  |  |
| --- | --- | --- |
| 用户服务类 | | |
| 类的描述：对Controller类传入数据进行处理 | | |
| 操作描述 | login () | 登录验证处理 |
| register() | 注册处理 |
| fixInfo () | 修改个人信息处理 |
| passwordFix() | 密码修改处理 |
| getUserInfoByUid() | 通过uid获取用户信息 |

用户数据访问类描述如下表6-3所示：

表6-3 用户数据访问类描述

|  |  |  |
| --- | --- | --- |
| 用户数据访问类 | | |
| 类的描述：对数据库进行CRUD操作 | | |
| 操作描述 | login () | 查找用户信息 |
| register() | 插入用户信息 |
| fixInfo () | 修改个人信息 |
| passwordFix() | 密码修改 |

用户实体类描述如下表6-4所示：

表6-4 用户实体类描述

|  |  |  |
| --- | --- | --- |
| 用户实体类 | | |
| 类的描述：用户实体类 | | |
| 属性  描  述 | uid | 用户id |
| userName | 用户名 |
| password | 密码 |
| sex | 性别 |
| age | 年龄 |
| tel | 手机号 |
| email | 邮箱 |
| detail | 详情 |

6.2.1.3 顺序图设计

用户登陆顺序图如下图6-4所示：

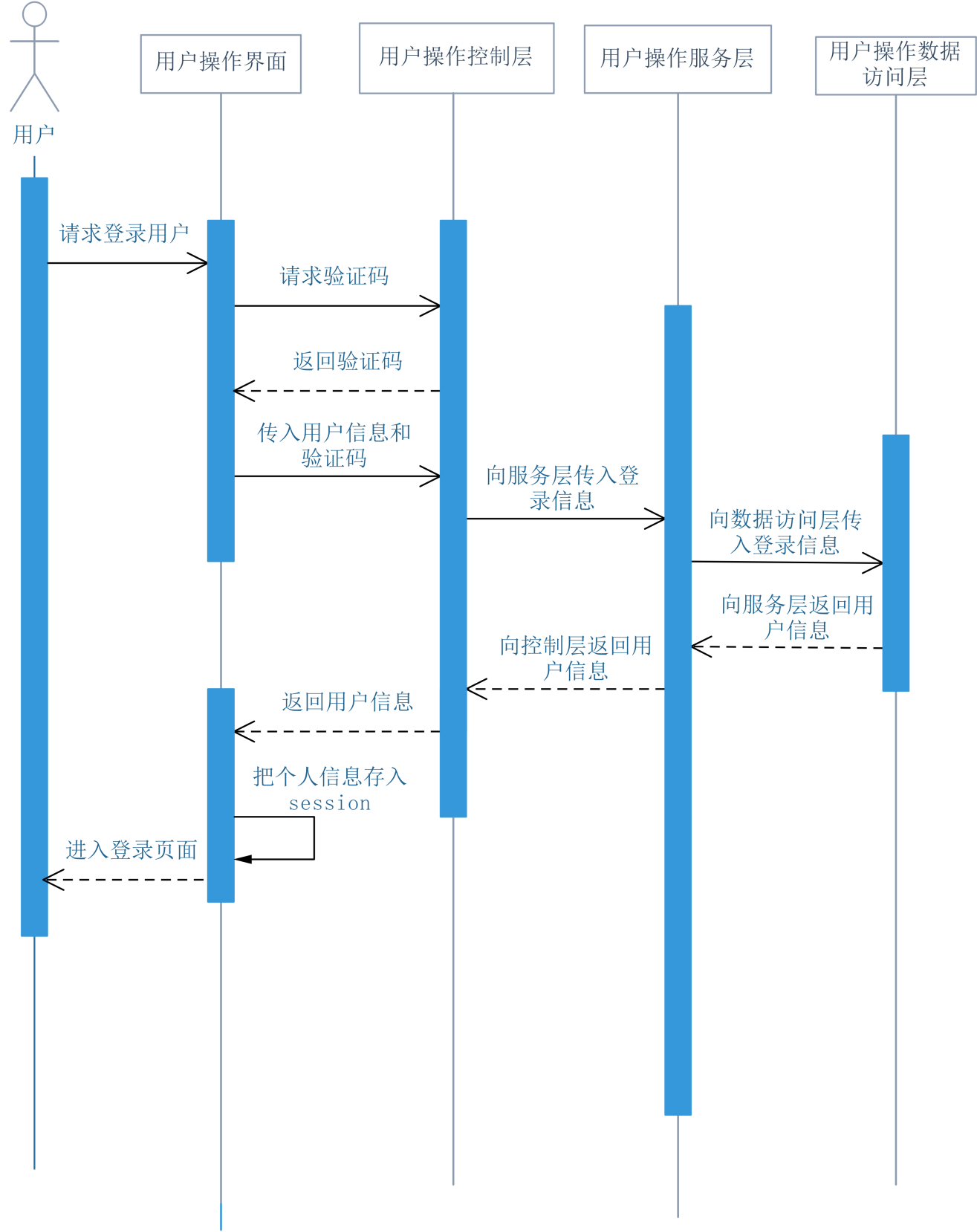


图6-4 用户登录顺序图

### 6.2.2信息发布

6.2.2.1功能结构设计

信息发布包图如下图6-5所示：

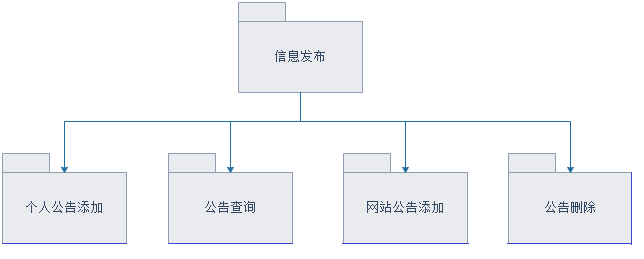


图6-5 信息发布包图

信息发布子系统包含“网站公告”与“个人公告”两个模块，在网站公告页面，用户只能查看公告，管理员将在管理员页面进行公告的添加、删除；在个人公告页面，用户可以查看到其他人发布的最新公告，也可以点击创建自己的公告。

6.2.2.2类图设计

公告控制类图如下图6-6所示：

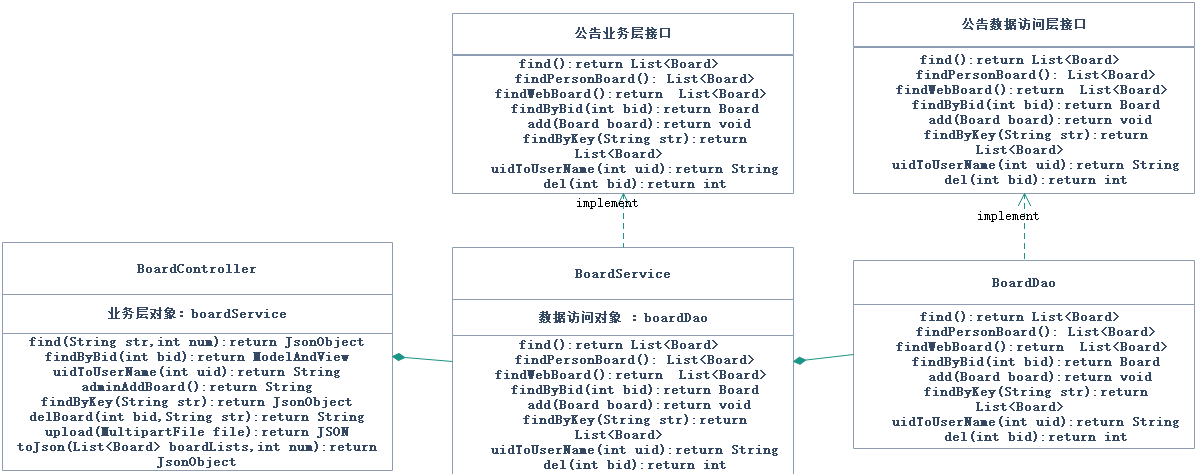


图6-6 公告控制类图

公告控制类描述如下表6-5所示：

表6-5 用户控制类描述

|  |  |  |
| --- | --- | --- |
| 公告控制类 | | |
| 类的描述：负责公告的创建、查找、删除 | | |
| 操作描述 | find () | 查找所有公告 |
| add() | 添加公告 |
| findByBid() | 通过id查询 |
| adminAdd () | 管理员添加 |
| findMyBoard() | 查找我的公告 |
| del() | 删除公告 |
| findByKey() | 通过关键字查询 |

公告服务类描述如下表6-6所示：

表6-6 公告服务类描述

|  |  |  |
| --- | --- | --- |
| 公告服务类 | | |
| 类的描述：负责对Controller传入的数据进行处理 | | |
| 操作描述 | find () | 查找所有公告 |
| add() | 添加公告 |
| findByBid() | 通过id查询 |
| adminAdd () | 管理员添加 |
|  | del() | 删除公告 |
| findByKey() | 通过关键字查询 |
|  | findMyBoard() | 查询我的公告 |

公告数据访问类描述如下表6-7所示：

表6-7 公告数据访问类描述

|  |  |  |
| --- | --- | --- |
| 公告数据访问类 | | |
| 类的描述：对数据库进行CRUD操作 | | |
| 操作描述 | find () | 查找所有公告 |
| add() | 添加公告 |
| del() | 删除公告 |
| findByBid() | 通过id查询 |
| adminAdd () | 管理员添加 |

公告实体类描述如下表6-8所示：

表6-8 公告实体类描述

|  |  |  |
| --- | --- | --- |
| 公告实体类 | | |
| 类的描述：公告实体类 | | |
| 属性  描  述 | bid | 公告id |
| tittle | 标题 |
| content | 内容 |
| uid | 用户id |
| time | 发布时间 |

6.2.2.3顺序图设计

添加个人公告顺序图如下图6-7所示：

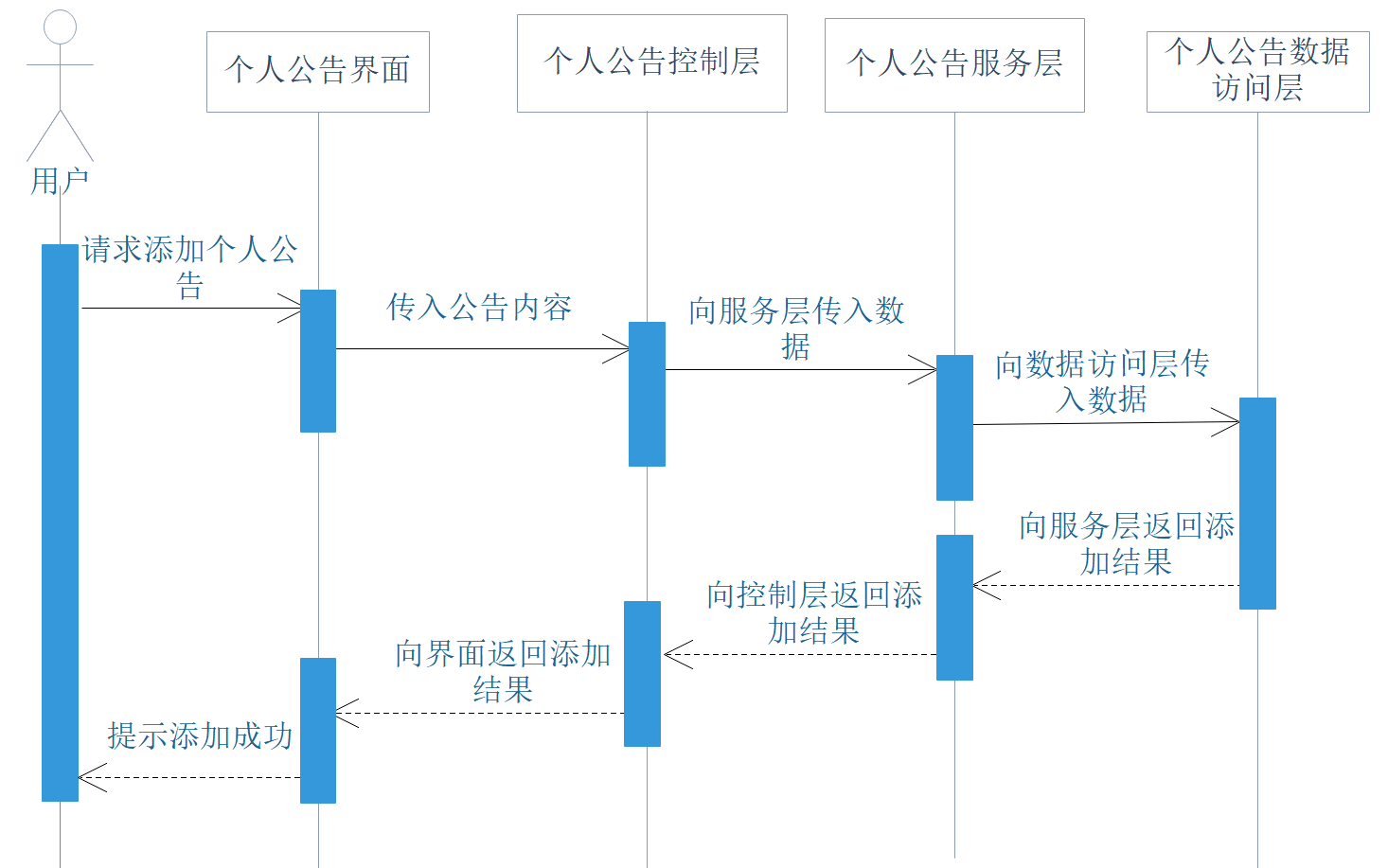


图6-7 添加个人公告顺序图

### 6.2.3咨询反馈

6.2.3.1功能结构设计

咨询反馈包图如下图6-8所示：

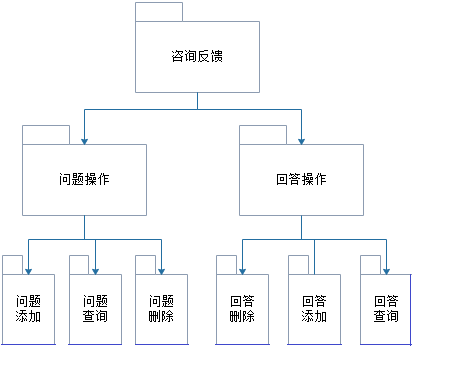


图6-8 咨询反馈包图

咨询反馈子系统拥有“问题”模块、“回答”模块，还有“管理员模块”，用户可以创建、查询问题，在问题详情界面可以对问题做出回答，点赞等操作，管理员则负责审核问题的合法性，回答的内容是否符合问题。对不合法的内容进行删除。

6.2.3.2类图设计

问题控制类图如下图6-9所示：



图6-9 问答控制类图

问答控制类描述如下表6-9所示：

表6-9 问答控制类描述

|  |  |  |
| --- | --- | --- |
| 问答控制类 | | |
| 类的描述：问题的管理、回答的管理 | | |
| 操作描述 | addQuestion () | 添加问题 |
| findAllQuestion() | 查找所有问题 |
| findQuestionByStr () | 通过关键字查询问题 |
| findQuestionByAid() | 通过id查询问题 |
| delAnswer() | 删除回答 |
| findMyAnswer() | 查找我的回答 |
| addAnswer() | 添加回答 |
| noAnswerQuestion() | 查找未回答的问题 |
| delQuestion() | 删除问题 |
| findMyQuestion() | 查找我的问题 |

问答服务类描述如下表6-10所示：

表6-10 问答服务类描述

|  |  |  |
| --- | --- | --- |
| 问答服务类 | | |
| 类的描述：对Controller类传入数据进行处理 | | |
| 操作描述 | addQuestion () | 添加问题 |
| findAllQuestion() | 查找所有问题 |
| findQuestionByStr () | 通过关键字查询问题 |
| findQuestionByAid() | 通过id查询问题 |
| delAnswer() | 删除回答 |
| findMyAnswer() | 查找我的回答 |
| addAnswer() | 添加回答 |
| noAnswerQuestion() | 查找未回答的问题 |
| delQuestion() | 删除问题 |
| findMyQuestion() | 查找我的问题 |

问答数据访问类描述如下表6-11所示：

表6-11 问答数据访问类描述

|  |  |  |
| --- | --- | --- |
| 问答数据访问类 | | |
| 类的描述：对数据库进行CRUD操作 | | |
| 操作描述 | addQuestion () | 添加问题 |
| findAllQuestion() | 查找所有问题 |
| findQuestionByStr () | 通过关键字查询问题 |
| findQuestionByAid() | 通过id查询问题 |
| delAnswer() | 删除回答 |
| findMyAnswer() | 查找我的回答 |
| addAnswer() | 添加回答 |
| noAnswerQuestion() | 查找未回答的问题 |
| delQuestion() | 删除问题 |
| findMyQuestion() | 查找我的问题 |

问题类描述如下表6-12所示：

表6-12 问题类描述

|  |  |  |
| --- | --- | --- |
| 问题类 | | |
| 类的描述：问题实体类 | | |
| 属性  描  述 | aid | 问题id |
| title | 标题 |
| content | 内容 |
| uid | 用户id |
| time | 发布时间 |

回答类描述如下表6-13所示：

表6-13 回答类描述

|  |  |  |
| --- | --- | --- |
| 回答类 | | |
| 类的描述：回答实体类 | | |
| 属性  描  述 | rid | 回答id |
| aid | 问题id |
| uid | 用户id |
| content | 内容 |
| answerTime | 回答时间 |

6.2.3.3顺序图设计

添加提问顺序图如下图6-10所示：

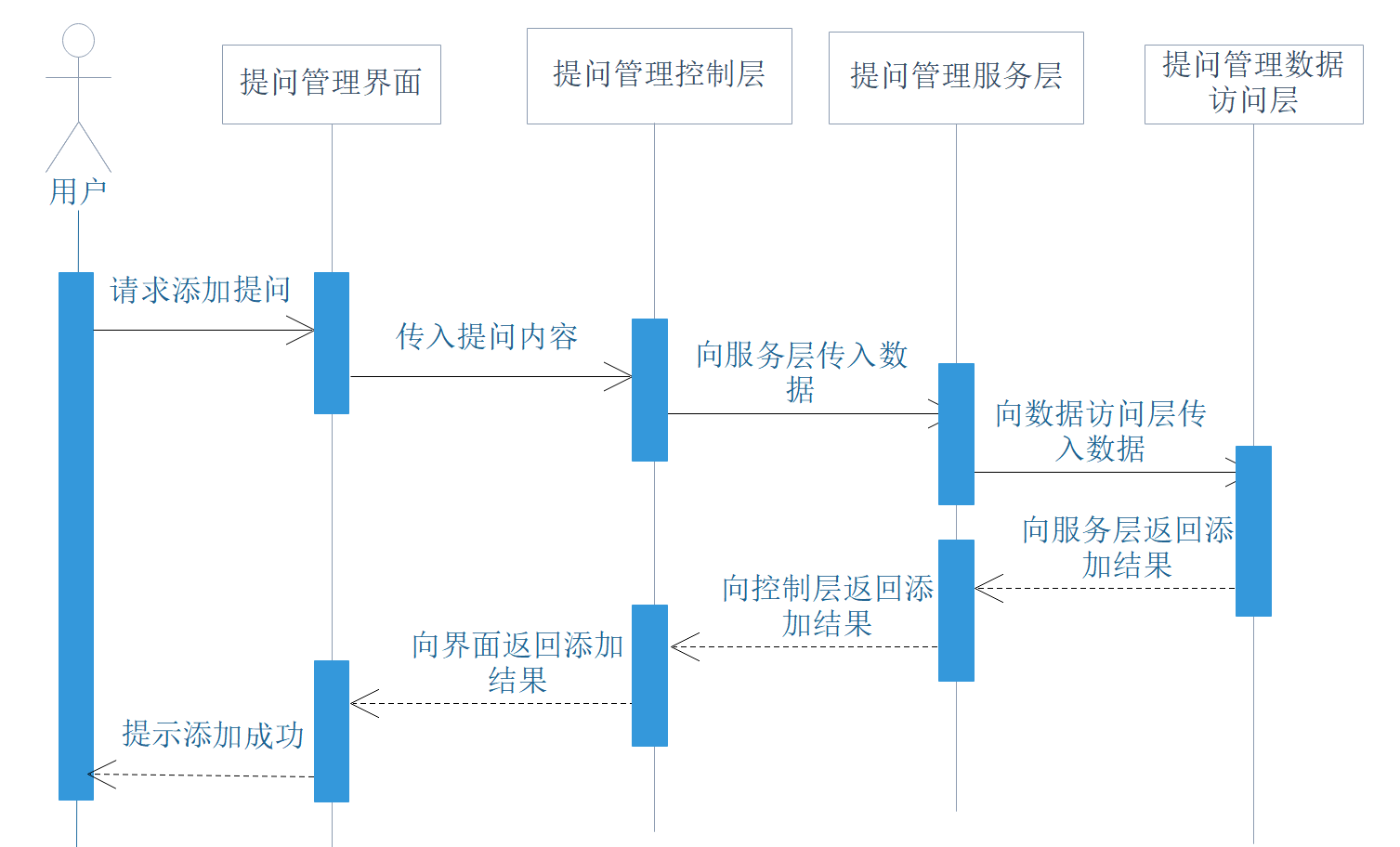


图6-10 添加提问顺序图

### 6.2.4线下研修室

6.2.4.1功能结构设计

线下研修室管理包图如下图6-11所示：

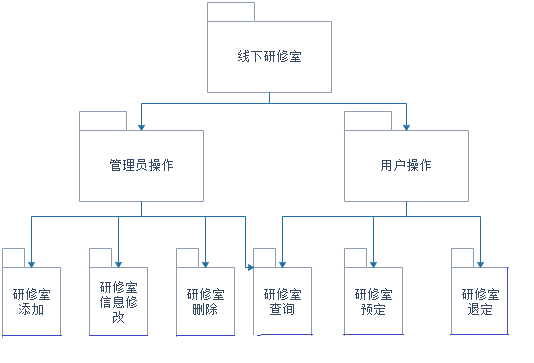


图6-11 线下研修室管理包图

线下研修室子系统包含“用户操作模块”，“管理员操作模块”，用户可以查询未被预定的子研修室，并进行预定，管理员则负责研修室的增加、删除、修改。

6.2.4.2类图设计

线下研修室控制类如下图6-12所示：

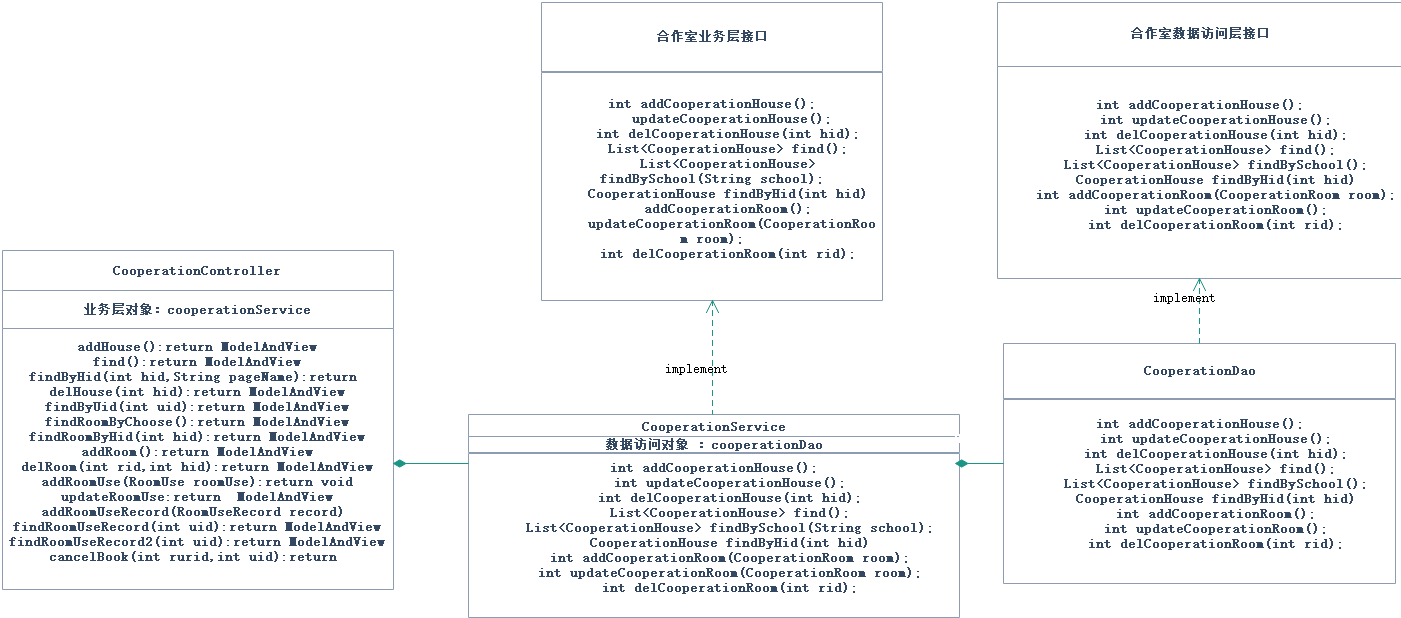


图6-12 线下研修室控制类图

线下研修室控制类描述如下表6-14所示：

表6-14 研修室控制类描述

|  |  |  |
| --- | --- | --- |
| 研修室控制类 | | |
| 类的描述：具有研修室的创建、查找、删除，预订、退订等功能 | | |
| 操作描述 | addHouse () | 添加研修室 |
| find() | 查找所有研修室 |
| findByHid() | 通过id查询研修室 |
| delHouse () | 删除研修室 |
| findByUid() | 通过用户id查询 |
| updateHouse() | 更新研修室信息 |
| findRoomByChoose() | 条件查询子研修室 |
| findRoomByHid() | 通过研修室id查询子研修室 |
| addRoom() | 添加子研修室 |
| delRoom() | 删除子研修室 |
| addRoomUse() | 添加房间使用情况表 |
| updateRoomUse() | 更新房间使用状态 |
| addRoomUseRecord() | 添加使用记录 |
| findRoomUseRecord() | 查询使用记录 |
| cancelBook() | 退订子研修室 |

研修室服务类描述如下表6-15所示：

表6-15 研修室服务类描述

|  |  |  |
| --- | --- | --- |
| 研修室服务类 | | |
| 类的描述：负责对Controller传入的数据进行处理 | | |
| 操作描述 | addHouse () | 添加研修室 |
| find() | 查找所有研修室 |
| findByHid() | 通过id查询研修室 |
| delHouse () | 删除研修室 |
| findByUid() | 通过用户id查询 |
| updateHouse() | 更新研修室信息 |
| findRoomByChoose() | 条件查询子研修室 |
| findRoomByHid() | 通过研修室id查询子研修室 |

表6-15 研修室服务类描述（续）

|  |  |  |
| --- | --- | --- |
|  | addRoom() | 添加子研修室 |
| delRoom() | 删除子研修室 |
| addRoomUse() | 添加房间使用情况表 |
| updateRoomUse() | 更新房间使用状态 |
| addRoomUseRecord() | 添加使用记录 |
| findRoomUseRecord() | 查询使用记录 |
| cancelBook() | 退订子研修室 |

研修室数据访问类描述如下表6-16所示：

表6-16 研修室数据访问类描述

|  |  |  |
| --- | --- | --- |
| 研修室数据访问类 | | |
| 类的描述：对数据库进行CRUD操作 | | |
| 操作描述 | addHouse () | 添加研修室 |
| find() | 查找所有研修室 |
| findByHid() | 通过id查询研修室 |
| delHouse () | 删除研修室 |
| findByUid() | 通过用户id查询 |
| updateHouse() | 更新研修室信息 |
| findRoomByChoose() | 条件查询子研修室 |
| findRoomByHid() | 通过研修室id查询子研修室 |
| addRoom() | 添加子研修室 |
| delRoom() | 删除子研修室 |
| addRoomUse() | 添加房间使用情况表 |
| updateRoomUse() | 更新房间使用状态 |
| addRoomUseRecord() | 添加使用记录 |
| findRoomUseRecord() | 查询使用记录 |
| cancelBook() | 退订子研修室 |

研修室实体类描述如下表6-17所示：

表6-17 研修室实体类描述

|  |  |  |
| --- | --- | --- |
| 研修室实体类 | | |
| 类的描述：研修室实体类 | | |
| 属性  描  述 | hid | 研修室id |
| school | 所属学校 |
| houseName | 研修室名称 |
| uid | 用户id |
| addTime | 添加时间 |
| tips | 备注 |

子研修室实体类描述如下表6-18所示：

表6-18 子研修室类描述

|  |  |  |
| --- | --- | --- |
| 子研修室类 | | |
| 类的描述：子研修室实体类 | | |
| 属性  描  述 | hid | 研修室id |
| rid | 子研修室id |
| major | 类型 |
| roomName | 子研修室名 |
| num | 容纳人数 |
| equipment | 设备 |
| tips | 备注 |

子研修室状态类描述如下表6-19所示：

表6-19 子研修室状态类描述

|  |  |  |
| --- | --- | --- |
| 子研修室状态类 | | |
| 类的描述：子研修室状状态类 | | |
| 属性  描  述 | ruid | 状态表id |
| rid | 子研修室id |
| time | 时间段 |
| state | 使用状态 |

子研修室使用记录类描述如下表6-20所示：

表6-20 子研修室状态类描述

|  |  |  |
| --- | --- | --- |
| 子研修室使用记录类 | | |
| 类的描述：子研修室使用记录类 | | |
| 属性  描  述 | ruid | 状态表id |
| rid | 子研修室id |
| uid | 用户id |
| time | 使用时间段 |
| addTime | 申请时间 |

6.2.4.3顺序图设计

修改研修室顺序图如下图6-13所示：

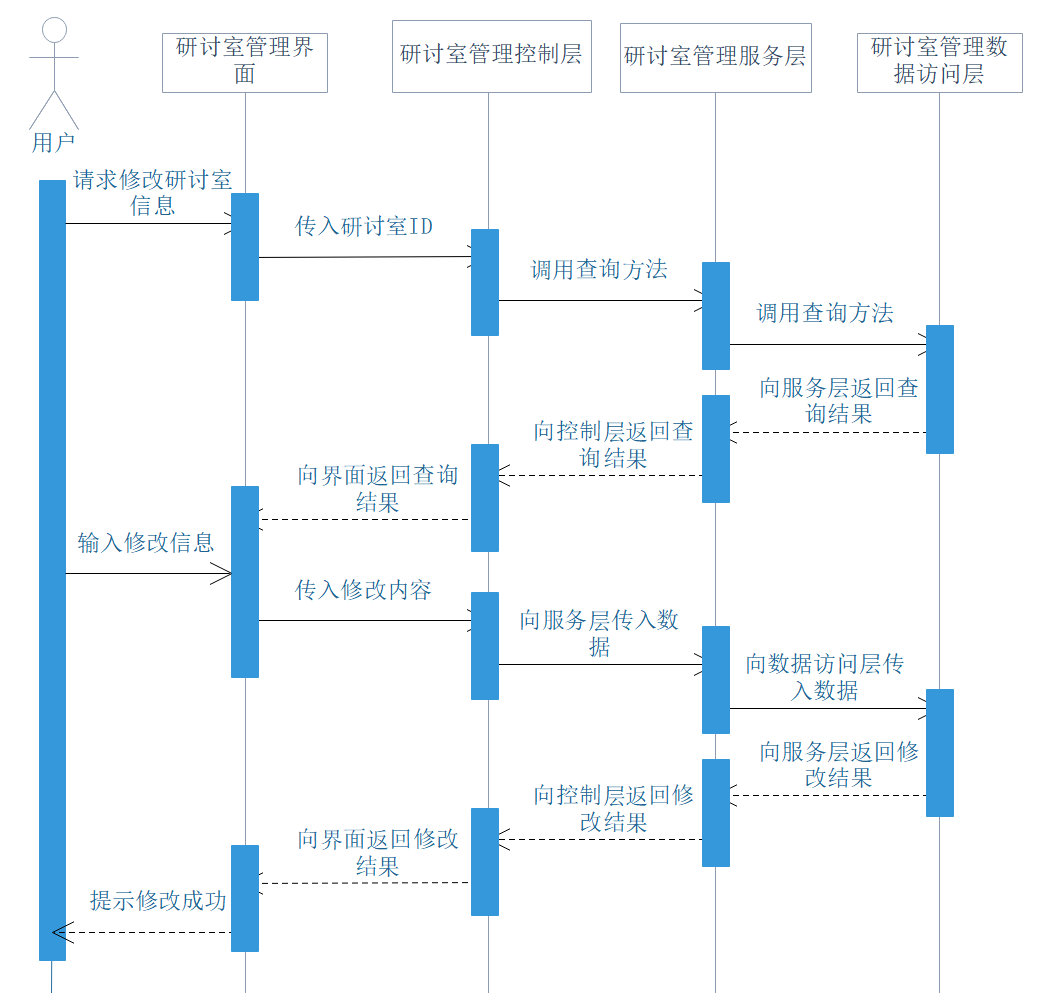


图6-13 修改研修室顺序图

7 编码

## 7.1 代码实现与核心算法

该控制层实现了对研修室的添加、查找所有的研修室、通过条件查询找出未被预定的研修室、增加子研修室、对子研修室的信息进行修改、查看某个用户的所有预定记录、查看用户预定的子研修室、将未到使用时间的预定子研修室进行退订操作、查看所有未被使用的子研修室等功能。核心代码实现如下所示：  
@RequestMapping("cooperation")  
public class CooperationController {  
  
 @Resource(name = "cooperationService")  
 private CooperationService cooperationService;  
  
 @RequestMapping("addHouse")  
 public ModelAndView addHouse(@RequestParam(name = "uid")int uid,  
 @RequestParam(name = "school")String school,  
 @RequestParam(name = "houseName")String houseName,  
 @RequestParam(name = "tips")String tips){  
 CooperationHouse cooperationHouse = new CooperationHouse(uid,school,houseName,tips);  
 cooperationService.addCooperationHouse(cooperationHouse);  
 return find();  
 }  
  
 @RequestMapping("find")  
 public ModelAndView find(){  
 List<CooperationHouse> cooperationHouseList = cooperationService.find();  
  
 ModelAndView mv = new ModelAndView();  
 mv.addObject("cooperationHouseList",cooperationHouseList);  
 mv.setViewName("adminHouse");  
 return mv;  
 }  
  
 @RequestMapping("findRoomByChoose")  
 @ResponseBody  
 public JSONObject findRoomByChoose(@RequestParam(name = "hid")int hid,  
 @RequestParam(name = "time")String time,  
 @RequestParam(name = "major")String major,  
 @RequestParam(name = "num")String num,  
 @RequestParam(name = "num1")int num1){  
  
 List<CooperationRoom> roomLists = cooperationService.findRoomByChoose(hid,time,major,num);  
  
  
 List<JSONObject> roomList = new ArrayList<>();//json房间表  
  
 List<List> roomUseLists = new ArrayList<>();//json时间表  
  
  
 for (int i = 10\*(num1-1);i < roomLists.size() && i < 10\*num1; i++) {  
 JSONObject object =new JSONObject();  
 List<RoomUse> ru = cooperationService.findByRid(roomLists.get(i).getRid());  
 List<String> timeList = new ArrayList<>();  
 for (int j = 0; j < ru.size(); j++) {  
 timeList.add(ru.get(j).getTime());  
 }  
 roomUseLists.add(timeList);  
 object.put("rid",roomLists.get(i).getRid());  
 object.put("roomName",roomLists.get(i).getRoomName());  
 object.put("major",roomLists.get(i).getMajor());  
 object.put("num",roomLists.get(i).getNum());  
 object.put("equipment",roomLists.get(i).getEquipment());  
 object.put("tips",roomLists.get(i).getTips());  
 roomList.add(object);  
 }  
  
 JSONObject jsonObject = toJson(roomLists,num1);  
 jsonObject.put("time",roomUseLists);  
 jsonObject.put("data",roomList);  
 return jsonObject;  
 }  
  
@RequestMapping("addRoom")  
 public ModelAndView addRoom(@RequestParam(name = "hid")int hid,  
 @RequestParam(name = "major")String major,  
 @RequestParam(name = "roomName")String roomName,  
 @RequestParam(name = "num")String num,  
 @RequestParam(name = "equipment")String equipment,  
 @RequestParam(name = "tips")String tips){  
 CooperationRoom room = new CooperationRoom(hid,major,roomName,num,equipment,tips);  
 cooperationService.addCooperationRoom(room);  
 int rid = room.getRid();  
 RoomUse roomUse = new RoomUse(rid,"上午");  
 this.addRoomUse(roomUse);  
 roomUse.setTime("下午");  
 this.addRoomUse(roomUse);  
 roomUse.setTime("晚上");  
 this.addRoomUse(roomUse);  
  
 return findRoomByHid(hid,"adminHouseDetail");  
 }  
  
 public JSONObject toJson2(List<RoomUseRecord> roomUseRecordLists, int num){  
 int page = roomUseRecordLists.size()/10;  
 if(roomUseRecordLists.size()%10 != 0) page++;  
 List<Object> js = new ArrayList<>();  
 for (int i = 10\*(num-1);i < roomUseRecordLists.size() && i < 10\*num; i++) {  
 JSONObject object =new JSONObject();  
 String roomName = cooperationService.findRoomByRid(roomUseRecordLists.get(i).getRid()).getRoomName();  
 object.put("roomName",roomName);  
 object.put("time",roomUseRecordLists.get(i).getTime());  
 object.put("addTime",roomUseRecordLists.get(i).getAddTime().toString());  
 js.add(object);  
 }  
  
 JSONObject jsonObject = new JSONObject();  
 jsonObject.put("data",js);  
 jsonObject.put("code", 0);  
 jsonObject.put("msg", "");  
 jsonObject.put("count", roomUseRecordLists.size());//数据条数  
 jsonObject.put("page",page);//总共几页  
 jsonObject.put("num",num);//当前页  
  
 return jsonObject;  
 }  
  
 @RequestMapping("findRoomUseRecord2")  
 public ModelAndView findRoomUseRecord2(@RequestParam(name = "uid")int uid){  
 List<RoomUseRecord> recordList = cooperationService.findRoomUseRecord2(uid);  
 Calendar cal=Calendar.getInstance();  
 int currentTime = cal.get(Calendar.HOUR\_OF\_DAY);  
  
 for (int i = 0; i < recordList.size(); i++) {//下午不能取消上午的预约  
 if(recordList.get(i).getTime().equals("上午")){  
 if(currentTime >= 8){  
 recordList.remove(i);  
 i--;  
 }  
 }else if(recordList.get(i).getTime().equals("下午")){  
 if(currentTime >= 14){  
 recordList.remove(i);  
 i--;  
 }  
 }else{  
 if(currentTime >= 19){  
 recordList.remove(i);  
 i--;  
 }  
 }  
 }  
 String[] roomNameList = new String[recordList.size()];  
 ModelAndView mv = new ModelAndView();  
 for (int i = 0; i < recordList.size(); i++) {  
 roomNameList[i] = cooperationService.findRoomByRid(recordList.get(i).getRid()).getRoomName();  
 }  
  
 mv.addObject("recordList",recordList);  
 mv.addObject("roomNameList",roomNameList);  
 mv.setViewName("myRoomUseRecord2");  
 return mv;  
 }  
}

## 7.2 代码优化

最开始我是用ModelAndView进行页面传值，如下图7-1，但用了一段时间后发现在前端传递变量时如果变量为二维数组，甚至多个二维数组，这样的方法使用起来就变得极为不便，并且我发现数据进行分页也很难通过这种方法实现，所以之后我采用了Layui的分页方法，如下图7-2、7-3、7-4，通过AJAX向后台传输数据，后台向前台传输JSON字符串，这样传输的变量就非常有序易懂，代码也更加易于维护。

<div style="background-color:antiquewhite;width: 65%;height: 80%;margin:20px 30px 30px;float: right">  
 <div class="layui-form">  
 <table class="layui-table">  
 <colgroup>  
 <col width="20%">  
 <col width="30%">  
 <col width="50%">  
 </colgroup>  
 <thead>  
 <tr>  
 <th>名称</th>  
 <th>所在大学</th>  
 <th>简介</th>  
 </tr>  
 </thead>  
 <tbody>  
 <c:forEach items="${cooperationHouseList}" var="coList">  
 <tr>  
 <th><a href="<%=session.getAttribute("basePath")%>/cooperationHouseDetail.jsp?hid=${coList.hid}">  
 <c:out value="${coList.houseName}"></c:out></a></th>  
 <th><c:out value="${coList.school}"></c:out></th>  
 <th><c:out value="${coList.tips}"></c:out></th>  
 </tr>  
 </c:forEach>  
 </tbody>  
 </table>  
 </div>  
</div>

layui.use(['laypage', 'form'], function () {  
 var laypage = layui.laypage  
 ,form = layui.form;  
  
 window.pageJump = function(num1) {  
 var str ="";  
 var basePath = "${sessionScope.get("basePath")}";  
 $.ajax({  
 url: basePath+'/cooperation/findRoomByChoose',  
 type : "post",  
 datatype : "json",  
 data:{"hid":<%=hid%>,"major":$('select[name="major"]').val()  
 ,"num":$('select[name="num"]').val(),  
 "time":$('select[name="time"]').val(),"num1":num1},  
 success: function (data) {  
 var jsObject = JSON.parse(data);  
 //alert(data);  
 var mydata = jsObject.data;  
 var timeList = jsObject.time;  
 $("#tbody").empty();  
 for (var i = 0; i < mydata.length; i++) {  
 var info = mydata[i];  
 var time = "\""+timeList[i]+"\"";  
 str =  
 '<tr>'+  
 '<th>'+info.roomName+'</th>'+  
 '<th>'+info.major+'</th>'+  
 '<th>'+info.num+'</th>'+  
 ' <th>'+info.equipment+'</th>'+  
 '<th>'+info.tips+'</th>'+  
 '<th>'+  
 '<button type="button" class="layui-btn layui-btn-normal" onclick='+"add("+info.rid+','+time+')>预定</button>'+  
 '</th>'+  
 '</tr>'  
  
 $("#tbody").append(str);  
 }  
  
 laypage.render({  
 elem:"demo1"  
 ,limit:10  
 ,pages:jsObject.page  
 ,count:jsObject.count  
 ,theme: '#1E9FFF'  
 ,curr:num1  
 ,jump: function (obj,first) {  
 num1 = obj.curr;  
 if(!first) {  
 pageJump(obj.curr);  
 }  
 }  
 })  
 form.render();  
 }  
 })  
 }  
 window.onload = pageJump(1);  
  
})

在进行分页的时候，开始我是采用物理分页，如下图7-5，这样能够了解数据库中有多少数据，并且通过拆解，获取到对应页的数据列表，但是一旦数据量偏大就会导致查询速度变慢，所以我有采用了逻辑分页，如下图7-6，从数据库中只取出对应页的数据，但是这样要获取全部的数据就需要重新写一个方法进行查询。

@RequestMapping("findQuestionByAid")  
public ModelAndView findQuestionByAid(@RequestParam(name = "aid")int aid,  
 @RequestParam(name = "str")String str){  
 Question question = questionService.findQuestionByAid(aid);  
 List<Answer> answerList = questionService.findAnswerByAid(aid);  
 String askName = this.uidToUserName(question.getUid());  
 List<String> nameList = new ArrayList();  
 if(answerList.isEmpty()){  
 }else {  
 for (Answer answer : answerList  
 ) {  
 nameList.add(this.uidToUserName(answer.getUid()));  
 }  
 }  
 ModelAndView mv = new ModelAndView();  
 mv.addObject("question",question);  
 mv.addObject("askName",askName);  
 mv.addObject("answerList",answerList);  
 mv.addObject("nameList",nameList);  
 mv.setViewName(str);  
 return mv;  
}

public JSONObject createJson(List<Question> askLists,int num){  
  
 int page = askLists.size()/10;  
 if(askLists.size()%10 != 0) page++;  
  
 List<JSONObject> askList = new ArrayList();  
  
 for (int i = 10\*(num-1);i < askLists.size() && i < 10\*num; i++) {  
 JSONObject object =new JSONObject();  
 object.put("aid",askLists.get(i).getAid());  
 object.put("title",askLists.get(i).getTitle());  
 object.put("time",askLists.get(i).getTime().toString());  
 askList.add(object);  
 }  
  
 JSONObject jsonObject = new JSONObject();  
 jsonObject.put("count", askLists.size());  
 jsonObject.put("data", askList);  
 jsonObject.put("page",page);//总共几页  
 jsonObject.put("num",num);//当前页  
 jsonObject.put("code", 0);  
 jsonObject.put("msg", "");  
 return jsonObject;  
}

# **8 测试**

## 8.1 测试方案设计

### 8.1.1 测试策略

功能测试策略如表8-1所示：

表 8-1 功能测试策略表

|  |  |
| --- | --- |
| 测试目标 | 确保测试的功能正常运行，在页面显示的数据也能够正常地排版、显示出来。对于一些风险具有较好的防范能力。 |
| 测试范围 | 公告模块（公告的添加功能、查询功能、管理员对公告的删除功能）,问题模块（问题的添加功能、查询功能、删除功能），回答模块（回答的添加功能、查询功能、删除功能），研修室模块（管理员对研修室的添加功能、修改功能、查询功能、用户的预定功能、退定功能） |
| 技术 | 利用正常数据和错误或者为空的数据测试内容：  在使用正常数据时得到预期结果。  在使用错误或者为空的数据时进行内部处理并以弹窗的形式提醒用户。 |
| 开始标准 | 通过登录检测 |
| 完成标准 | 功能运行正常，对于非常规操作有一定的识别能力，没有高风险安全漏洞 |
| 测试重点和优先级 | 公告添加功能、用户退定研修室功能 |
| 需考虑的特殊事项 | 测试环境、网络问题、使用次数 |

系统测试策略如表8-2所示：

表 8-2 系统测试策略表

|  |  |
| --- | --- |
| 测试目标 | 检测需求中业务流程，数据流的正确性 |
| 测试范围 | 需求中明确的业务流程，或组合不同功能模块而形成一个大的功能 |
| 技术 | 利用有效的和无效的数据来执行各个功能，以核实一下内容： 在使用有效数据时得到预期结果。  在使用无效数据时显示错误消息或提示消息。  各业务规则都得到了正确的应用。 |
| 开始标准 | 在完成某个集成测试时必须达到标准 |
| 完成标准 | 已执行全部的计划测试  发现的高风险的缺陷已经全部解决 |
| 测试重点和优先级 | 研修室预定功能 |
| 需考虑的特殊事项 | 异地登陆、网络波动问题、网站登录时长问题 |

### 8.1.2 测试进度安排

测试进度安排如表8-3所示：

表 8-3 测试进度安排

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试阶段 | 测试任务 | 工作量估计 | 人员分配 | 起止时间 |
| 第一阶段功能测试 | 添加公告  问题操作  回答操作  研修室预定/退定功能 | 10 日 |  | 2020 年 4 月 1 日至  2020 年 4 月 7 日 |
| 第二阶段系统测试 | 对于一个模块一整套流程的测试  确保所有功能正常，数据显示正常 | 3 日 |  | 2020 年 4 月 9 日至  2020 年 4 月 12 日 |
| 第三阶段  性能测试 | 进行攻击测试、高并发测试 | 1 日 |  | 2020 年 4 月 18 日 |
| 第四阶段  安装部署测试 | 将项目部署到服务器进行测试 | 2 日 |  | 2020 年 4 月 19 日至  2020 年 4 月 20 日 |

### 8.1.3 测试资源

（1）测试环境

Windows 10 操作系统，2G 独立显卡，CPU 为 INTEL CORE i5 处理器，硬盘 500G，

内存 4G，Tomcat9，FireFox，Mysql8.0，Java 1.8。

（2）测试管理工具

FireFox：Firefox支持非常多的网络标准，如[标准通用标记语言](https://baike.baidu.com/item/%E6%A0%87%E5%87%86%E9%80%9A%E7%94%A8%E6%A0%87%E8%AE%B0%E8%AF%AD%E8%A8%80" \t "https://baike.baidu.com/item/Mozilla%20Firefox/_blank)下的子集[HTML](https://baike.baidu.com/item/HTML)和[XML](https://baike.baidu.com/item/XML" \t "https://baike.baidu.com/item/Mozilla%20Firefox/_blank)、[XHTML](https://baike.baidu.com/item/XHTML" \t "https://baike.baidu.com/item/Mozilla%20Firefox/_blank)、[SVG](https://baike.baidu.com/item/SVG" \t "https://baike.baidu.com/item/Mozilla%20Firefox/_blank) 1.1（部分的）、[CSS](https://baike.baidu.com/item/CSS" \t "https://baike.baidu.com/item/Mozilla%20Firefox/_blank)（除了标准之外，还有扩充的支持）、[ECMAScript](https://baike.baidu.com/item/ECMAScript" \t "https://baike.baidu.com/item/Mozilla%20Firefox/_blank)（[JavaScript](https://baike.baidu.com/item/JavaScript)）、[DOM](https://baike.baidu.com/item/DOM" \t "https://baike.baidu.com/item/Mozilla%20Firefox/_blank)、[MathML](https://baike.baidu.com/item/MathML" \t "https://baike.baidu.com/item/Mozilla%20Firefox/_blank)、[DTD](https://baike.baidu.com/item/DTD" \t "https://baike.baidu.com/item/Mozilla%20Firefox/_blank)、[XSLT](https://baike.baidu.com/item/XSLT" \t "https://baike.baidu.com/item/Mozilla%20Firefox/_blank)、[XPath](https://baike.baidu.com/item/XPath" \t "https://baike.baidu.com/item/Mozilla%20Firefox/_blank)和PNG图像文件（包含透明度支持）。

（3）测试文档

边界测试文档、页面测试文档、白盒测试文档

### 8.1.4 关键测试点

1.公告的创建时富文本的获取，以及最长内容的限定，对于脚本注入的防范测试。

2.咨询反馈子系统中对于输入内容及长度的限定测试，防止sql注入的测试。

## 8.2 测试用例构建

### 8.2.1 测试用例编写约定

用例编号： 唯一标识，与需求编号对应，为多对一关系。

用例名称： 对测试项简短的描述。

前提条件： 用例执行时需要的预置条件。

操作步骤： 该动作 执行需要完成的操作。

预期结果： 该动作执行完后程序的表现结果。

实际结果： 实际输出的结果。

问题描述： 执行该用例出现后系统显示的错误。

验证结果： 该测试用例是否执行通过。

BUG 编号： 记录出现该种状况的BUG编号。

测试结果与结论：记录测试结果是否与预想相同。

测试执行者：按照标准进行测试的人。

### 8.2.2 测试用例设计

用户登录测试用例如表8-4所示：

表 8-4 用户登录测试用例表

|  |  |
| --- | --- |
| 用例编号 | Cooperation 1-1 |
| 用例名称 | 用户登录 |
| 前置条件 | 用户进入系统登录界面 |
| 操作步骤 | 1）输入用户名、密码、验证码 |
| 预期结果 | 1）核对信息成功、验证码正确，进入系统首页 |
| 实际结果 | 与预期结果相符 |
| 问题描述 | 暂无 |
| BUG 编号 | 暂无 |
| 测试结果与结论 | 测试通过 |
| 测试执行者 |  |

用户添加公告测试用例如表8-5所示：

表 8-5用户添加公告测试用例表

|  |  |
| --- | --- |
| 用例编号 | Cooperation 1-2 |
| 用例名称 | 用户添加公告 |
| 前置条件 | 用户进入公告页 |
| 操作步骤 | 不输入标题，输入内容  上传图片点击其他格式文件 |
| 预期结果 | 标题输入框变红，光标自动跳至标题输入框  弹出提示“文件类型不符合” |
| 实际结果 | 与预期结果相符 |
| 问题描述 | 暂无 |
| BUG 编号 | 暂无 |
| 测试结果与结论 | 测试通过 |
| 测试执行者 |  |

### 8.2.3 关键测试用例

用户研修室预定测试用例如表8-6所示：

表 8-6 用户研修室预定测试用例表

|  |  |
| --- | --- |
| 用例编号 | Cooperation 1-3 |
| 用例名称 | 用户研修室预定 |
| 前置条件 | 用户进入研修室选择页 |
| 操作步骤 | 用户选择时间段后点击提交  用户不选择时间段直接点击提交  预定后，该研修室该时间段从预定时间中消失 |

表 8-6 用户研修室预定测试用例表（续）

|  |  |
| --- | --- |
| 预期结果 | 提交成功，个人信息页有预定记录  弹出提示框显示请输入预定时间段  点击预定，可选时间段变少 |
| 实际结果 | 与预期结果相符 |
| 问题描述 | 暂无 |
| BUG 编号 | 暂无 |
| 测试结果与结论 | 测试通过 |
| 测试执行者 |  |

### 8.2.4 测试用例维护

1. 每一次对功能的更新修改都要重新对该功能进行测试，并且思考是否产生了新的漏洞，要对漏洞即时修复。思考功能更新带来的一系列操作变动，尽量减小变动带来的影响。
2. 对于修改的功能，之前的测试方法也要随之进行改动，保证可以正常进行测试，但要将之前的测试用例移出项目，而不是直接删除，保证若需要版本回退时，能够正常测试。
3. 设计测试用例时应注意用例的兼容性，保证上线后可以根据用户所反映的问题，及时地处理设计不当的用例。若用例写得太绝对化，很容器出现用例修改困难，甚至牵连到其他用例的正常运行。
4. 根据新的需求文档，要设计出新的测试用例

9 总结与展望

## 9.1 设计工作总结

在这次毕业设计中我再次重新回顾了一下SpringMVC，了解了MVC架构，也知道了还有MVVM和MVP架构，知道了他们的不同使用场景。为了前端页面的美化，学会了Layui框架的一部分功能，熟悉了AJAX的内部属性及其作用，了解了JSON在前后台传输时必须通过序列化与反序列化才能得到对象。

我负责的公告子系统，功能基本实现，能够将公告按发布人员进行分类，并进行分页处理，显示最新的公告在最上方，预期未来添加公告点击排行在右侧。问答子系统仅完成了基本的提问，回答，查询，对于点赞只完成了前端部分，举报功能只有个雏形，能够实现，但未完善，没有经过测试，相应的最优回答奖励也没实现。用户子系统的个人信息处理基本完成，能够进行资料修改，密码修改，但用户之间未产生联系，无法进行快速用户查询，session过期问题未解决。线下研修室子系统功能完善，但缺少实地的管理，对于预定时间只能按固定时间段进行预定，无法按个人时间预定，分类不明确，还未形成详细的管理制度。

在功能实现过程中，我通过查询资料、博客等实现了很多原本不会的功能，从中我吸取了大量知识，了解并加深了很多方法的理解。每日的学习也保证了我不在假期当中变得懒散，保持着每日规律的作息时间。

## 9.2 未来工作展望

对该项目，今后可以添加好友功能，便于线上视频交流与文字交流的开展。解决当前无法解决的session过期问题，当过期后能够自动跳转到登录页面，而不是报错。公告能够显示的更加详细，完善举报功能后可进行线上举报，特定的维护人员进行处理，问答设置奖项，提高用户回答问题的积极性，同时对胡乱回答的内容进行删除，并对回答用户进行禁言等类似处罚。对于线下研修室，则需要现实中的各个部门进行分工合作，保证研修室真正地被需要使用的人使用，并且保证研修室内设施不会出现非正常损耗，以及卫生等一系列琐碎事情的处理。

将管理员细分，可初步按照子系统进行拆分，明确各类管理员的权限，保证线上线下维护的共同进行，后期运转中再根据实际情况处理。

未来可采用redis数据库，保证读写分离，加快查询速度，运用类似Token的技术使每个学校都有自己的数据库，大大提高网页访问速度。通过负载均衡的方法扩展[网络设备](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E8%AE%BE%E5%A4%87/7667828" \t "https://baike.baidu.com/item/%E8%B4%9F%E8%BD%BD%E5%9D%87%E8%A1%A1/_blank)和[服务器](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8/100571)的带宽、增加[吞吐量](https://baike.baidu.com/item/%E5%90%9E%E5%90%90%E9%87%8F/157092)、加强网络数据处理能力、提高网络的灵活性和可用性。

谢 辞

参考文献

1. 魏书寒.基于SSM框架的图书管理系统的设计与实现[J].工业控制计算机，2017（7）：133-134.
2. 周国华.基于SSM框架学习平台的设计与实现[J].信息与电脑，2017（24）：138-139.
3. 罗宏俊，冯瑞.基于Web技术进行移动应用开发和中间件的研究[J].计算机系统应用，2017.11：19-27.
4. 余楷鑫.Bootstrap在Web移动开发中的应用[J].电脑知识与技术，2017.6：82-84.
5. 郭冰.基于MySQL数据库的索引优化研究[J].信息与电脑（理论版），2019，（12）：154-156，163.
6. 赵骥，张东洋，刘业顺.基于MVC模式和Java EE技术仓储管理系统的设计与实现[J].无线互联科技，2019（6）.
7. 周佳雯，胡玉欣，周文华等.基于SSM框架的活动审批Web系统的设计与实现[J].微型电脑应用，2017（6）：50-60.
8. FAIN，RA. Enterprise Web development［M］. Sebastopol O，Reily Media，2017.
9. Liang B.The study and application of the new control layer for enterprise-class web applications [J].Journal of Computers，2017，28 （6）：151-162.
10. 王英英. Mysql从入门到精通［M］. 北京：清华大学出版社，2019. 08.
11. 李鸿征.协同创新驱动发展战略下的地方高校转型发展研究[J] 河南：河南省软科学研究计划项目，2018.3.17
12. 李刚.疯狂Java讲义[M] 北京：电子工业出版社，2020.03
13. 丁亚飞，薛燚. HTML5+CSS3+JavaScript案例实战[M] 北京：清华大学出版社，2020.01
14. Hugo Di Francesco, Siyuan Gao. Professional JavaScript: Fast-track your web development career using the powerful features of advanced JavaScript[M] UK：English Edition，2019.09
15. 吴为胜，杨章伟. Spring+Spring MVC+MyBatis从零开始学[M] 北京：清华大学出版社，2019.07
16. 谭丽娜，陈天真. Web前端开发技术（jQuery+AJAX）[M] 北京：人民邮电出版社，2019.07
17. 魏梦舒.漫画算法：小灰的算法之旅[M] 北京：电子工业出版社，2020.03
18. Amit Konar.Artificial Intelligence and Soft Computing: Behavioral and Cognitive Modeling of the Human Brain[M] UK：English Edition，2018.10
19. [虞卫东](https://www.amazon.cn/s/ref=dp_byline_sr_ebooks_1?ie=UTF8&field-author=%E8%99%9E%E5%8D%AB%E4%B8%9C&text=%E8%99%9E%E5%8D%AB%E4%B8%9C&sort=relevancerank&search-alias=digital-text).深入浅出HTTPS：从原理到实战[M] 北京：电子工业出版社，2018.06
20. 高洪岩.NIO与Socket编程技术指南(Java核心技术系列)[M] 北京：机械工业出版社，2018.07
21. [段钢](https://book.douban.com/search/%E6%AE%B5%E9%92%A2).加密与解密（第4版）[M].北京：电子工业出版社，2018.
22. [软件过程与管理方法综述](https://x.cnki.net/kcms/detail/detail.aspx?filename=RJXB201901005&dbcode=CJFQ&dbname=CJFDTEMP&v=)[J]. 荣国平,张贺,邵栋,王青.  软件学报. 2019(01)
23. [新型数据管理系统研究进展与趋势](https://x.cnki.net/kcms/detail/detail.aspx?filename=RJXB201901010&dbcode=CJFQ&dbname=CJFDTEMP&v=)[J]. 崔斌,高军,童咏昕,许建秋,张东祥,邹磊.  软件学报. 2019(01)
24. [大数据管理系统的历史、现状与未来](https://x.cnki.net/kcms/detail/detail.aspx?filename=RJXB201901008&dbcode=CJFQ&dbname=CJFDTEMP&v=)[J]. 杜小勇,卢卫,张峰.  软件学报. 2019(01)
25. [基于内容的教育调查问卷系统的设计与实现](https://x.cnki.net/kcms/detail/detail.aspx?filename=1018288691.nh&dbcode=CMFD&dbname=CMFDTEMP&v=)[D]. 石河.吉林大学 2018
26. [基于SSM框架的通用电子问卷调查系统设计与实现](https://x.cnki.net/kcms/detail/detail.aspx?filename=1018087300.nh&dbcode=CMFD&dbname=CMFDTEMP&v=)[D]. 张玮.安徽大学 2018
27. [设置安全的SQL Server](https://x.cnki.net/kcms/detail/detail.aspx?filename=WAXX201903040&dbcode=CJFQ&dbname=CJFD2019&v=)[J]. 左振辉,陆世炜.  网络安全和信息化. 2019(03)
28. [软件测试质量评价方法](https://x.cnki.net/kcms/detail/detail.aspx?filename=JYXH201809010&dbcode=CJFQ&dbname=CJFDTEMP&v=)[J]. 李军锋,顾滨兵,李海浩.  计算机与现代化. 2018(09)
29. [基于非结构化文本的自动问答系统的研究与实现](https://x.cnki.net/kcms/detail/detail.aspx?filename=1019114157.nh&dbcode=CMFD&dbname=CMFD2019&v=)[D]. 刘庆.北京邮电大学 2019
30. [研究生招生咨询智能问答系统的设计与实现](https://x.cnki.net/kcms/detail/detail.aspx?filename=1019114188.nh&dbcode=CMFD&dbname=CMFD2019&v=)[D]. 丁怡心.北京邮电大学 2019
31. [面向阅读理解任务的问答系统研究与实现](https://x.cnki.net/kcms/detail/detail.aspx?filename=1019112744.nh&dbcode=CMFD&dbname=CMFD2019&v=)[D]. 刘珮.北京邮电大学 2019
32. [基于阅读理解的知识问答系统设计与实现](https://x.cnki.net/kcms/detail/detail.aspx?filename=1018708879.nh&dbcode=CMFD&dbname=CMFD2018&v=)[D]. 张帅帅.电子科技大学 2018

附录A 外文翻译—原文部分

出自《WEBRTC权威指南》机械工业出版社 第三版

Important step is to establish a peer-to-peer connection using the corresponding API. RTCPeerConnection APT is the core of WebRTC and, as the name implies, is used to establish a connection between two peers. In this book, "peer" refers to two communication terminals on the world wide web, synonymous with peers in the term "peer file sharing." This is not through the server requesting communication, but directly between the two entities. In WebRTC itself, a peer-to-peer connection is a direct media connection between two Web browsers. This model is very useful, especially for multidirectional communication, such as teleconferencing between three or more browsers. Each pair of browsers needs a peer-to-peer connection to join the meeting, so the audio and video streams can flow directly between the two browsers, as shown in figure 1.10. Therefore, if you have three browsers communicating, you will need to establish a total of three connections. The application developer will need to establish a peer-to-peer connection for each pair of browsers to connect to, or browsers to, another terminal, such as an existing communications network. You can also use another architecture as shown in figure 1.11.  
 A new RTCPeerConnection object is required to establish this connection. The only input to the RTCPeerConnection constructor method is a configuration object that contains the information that ICE (Interactive Connectivity Establishment) (see section 10.2.7) USES to "punch holes" through various Network AddressTranslation (NAT) devices and firewalls.

MediastreamTrack is the basic media unit in WebRTC. This track represents a single type of media that a device or recorded content (called a "source") can return. A single stereo source or a 6-channel surround sound signal can be considered a single track, although both consist of multiple audio channels. Note that the specification does not define how the media can be accessed or manipulated at the channel level, although it roughly states that the channels "have an accepted relationship with each other". In fact, as defined in the WebRTC documentation, the contents of the tracks "will be encoded together for transmission as some payload type, such as RTP". In other words, when a peer-to-peer connection is used for transmission, the tracks are treated as a unit, even if they are locally enabled/disabled or silent.  
 Each orbital has a source associated with it. We will explain this later, but for now it is clear that there is no direct access or control of the source through WebRTC. All control of the source is exercised through the track. Currently, the relationship between MediaStreamTrack and its sources is changing somewhat, but in the following direction: the track can be not only the original media from the source, but also the converted version provided by the browser. For example, the track could represent a low-sampling rate version of the video recorded by the camera at a higher native resolution.

MediaStream is a collection of MediaStreamTrack objects. There are two ways to create these Mediastream objects: one is to request access to local media by copying tracks from existing MediaSream; The second is to use peer-to-peer connections to receive new streams. Currently, there is only one way to request and access this media, which is through a call to getUserMedia(), but other methods, such as streaming from a local file, may appear in the future. Tracks of existing MediaStream objects can be "copied" into new MediaSream objects through the MediaStream constructor, which takes an existing MediaStream object or a series of MediaStreamTrack objects as parameters. You can also specify no parameters at all and add tracks (addTrack) and remove tracks (removeTrack) to existing MediaSream, but this is not yet widely supported. In addition, if you don't like passing MediaStream to the constructor, you can use the clone) method to copy the stream (and all its tracks).

In real-time communication, the main role of signaling is reflected in four aspects :

1) negotiate media functions and Settings.  
 2) identify and verify the identity of session participants.  
 3) control the media session, indicate progress, change the session, and terminate the session.  
 4) dual occupancy Resolution is implemented when both parties of the session attempt to establish or change the session at the same time.  
The next few sections will explore these capabilities in detail, explaining why item 1 is an essential feature in WebRTC, how it is standardized, and why items 2, 3,and 4 are optional features or just part of a Web application.

When a real-time communication is initiated using a standard signaling protocol, such as SIP or Jingle, the signaling channel provides the identity of the participant and the option to authenticate. In WebRTC, there are two channels available for identification in addition to signaling. The first is to see the context provided by the Web application. For example, a user of the WebRTC web site might log in with a specific screen name. When this user wishes to establish a conversation with another user, the Web application provides the other user with its screen name as an identity, leaving the user on the other end with no choice but to believe that the site provides an accurate identity. This is very similar to the caller id in the Public Switched Telephone Network (PSTN). PSTN users can only believe that the caller ID presented to their phone by their service provider is the caller, and there is no other way to independe.

HTTP polling is a simple proprietary signaling scheme. For example, the xhr-based signaling channel described later in this chapter is HTTP polling.  
 By calling XHR in JavaScript or jQuery, you can enable JavaScript applications to generate new HTTP requests against the Web server and process HTTP responses. XHR is a w3C standard API currently defined as a working draft [XHR], and XHR is on its way to becoming a standard. Each functional component of XHR JavaScript API is supported by the browser. Although XHR only contains HTTP requests in its name, in addition to sending XML requests, it can also send JavaScript Object Notation (ISON) or plaintext. XHR enables the browser to generate new HTTP or HTTPS requests, such as GET, POST,PUT, and so on. The corresponding API call specifies the method to use along with the IP address and port number. The response to the request is returned to JavaScript  
 To use XHR as the WebRTC signaling channel, the Web server needs to run an application to receive HTTP requests over another XHR channel and to proxy or forward information received from one browser to another browser, as shown in figure 4.7.  
To exchange signaling information, the JavaScript running in each browser periodically sends HTTP messages to the signaling server for polling. The signaling information sent by the browser is contained in the POST method, and the signaling information received by the server is contained in the 200 0K response sent to the POST. Note that in this scenario, each message is a new request.  
 WebRTC allows the browser to use any method of communication to transmit signaling information, from smoke signals to Tin Can with strings, to faster and more common solutions such as HTTP (XML HTTP requests), WebSocket, and Google applications. These different scenarios have been explored and compared earlier in this chapter. Note, however, that in any case, the browser is communicating with the Web server, not the browser on the other side. While browsers are not restricted from communicating directly with each other, the Web model itself assumes that the browser (the Web client) only communicates with the server and not directly with other Web clients. In fact, Web clients have no idea how to communicate with each other unless the Web server provides the IP address of the Web client. Thus, in most cases, there is a server in the cloud that ACTS as a relay for sending messages from one browser to another.  
 Generally speaking, Web server-based signaling schemes fall into one of two categories: polling schemes and session-oriented schemes. In a polling scenario represented by XHR, all communication is initiated by the browser, and the only way for the browser to receive a message from the Web server is to receive the server's response to its message. In order to receive all messages from the server, the browser needs to periodically send messages to the server for inquiry in case there are messages waiting to be received. In a session-oriented scenario, such as WebSocket, a virtual connection is established between the browser and the web server to send messages to each other.

To simplify application programming, it is convenient to define a signaling channel interface that exposes the connect and send methods and can specify the handler for the message's time of receipt. If the application is to use WebRTC in peer mode (" triangle "mode), the role of the virtual signaling channel is to provide a simple connection-oriented model for sending signaling information to peers. There are several issues to be aware of when creating this virtual connection.Send messages to each other.  
 The first, and most important, is to determine how to indicate to the Web server the nature of the connection to be established. Is this connection really just for one peer, or is it to set up a group "call"? Given that it is a one-to-one connection, how do you identify the other party? A reasonable solution is to have each browser registered with the server and have the server send back a list of registered browsers so that the user can choose which browser to communicate with. Another solution is to use the key that the browser on the other end knows and provides. The demo code in this chapter USES the latter scenario. In this particular demonstration, the first browser connected with the given key registers as the first party and is told to wait. When the second client connects using the same key, a signaling channel is established between the two clients. Later, if a new party tries to make another connection with the same key, the existing connection is destroyed and instructed to wait. Next, when another browser connects with the same key, a signaling channel is built between the two, and so on, its state shifts between waiting and connected. The potential downside of this scenario is that both parties need to know their keys and the new party needs to destroy the connection. Still, it has two very beneficial features; The key can be included as a parameter in the URI, so simply add the URI as a bookmark to save the "connection," and when the connection is  a browser or network failure disconnects, very little state needs to be managed, and when the connection is broken, both parties simply reload their pages to resume communication.

附录B 外文翻译—译文部分

一个重要步骤是使用相应的API建立对等连接。RTCPeerConnection APT是WebRTC的核心,顾名思义,其作用就是在两个对等端之间建立连接。在本书中,“对等端”是指万维网上的两个通信终端,与术语“对等端文件共享”中的对等端同义。这不是通过服务器请求通信,而是直接在两个实体之间进行通信。就WebRTC本身而言,对等连接就是两个Web浏览器之间的直接媒体连接。这种模式非常实用,尤其适用于多向通信,例如在三个或更多浏览器之间建立的电话会议。每一对浏览器都需要一个对等连接才能加入会议,这样音频和视频流可直接在两个浏览器之间流动,如图1.10所示。因此,如果有三个浏览器进行通信,总共将需要建立三个连接。应用程序开发人员将需要为每一对要连接的浏览器或浏览器与其他终端,例如现有通信网络)建立一个对等连接。也可以采用图1.11中所示的另一种体系结构。  
 建立此连接需要一个新的RTCPeerConnection对象，RTCPeerConnection构造函数方法的。

唯一输入是一个配置对象，该对象包含IEC（Interactive Connectivity Establishment,交互式连接建立技术）“打洞”通过各种网络地址转换(Network AddressTranslation，NAT)设备和防火墙时所使用的信息。

MediastreamTrack是WebRTC中的基本媒体单元。此轨道代表一种设备或录制内容(称为“源”)可返回的单一类型的媒体。单个立体声源或6声道环绕声音频信号均可视为一个轨道，尽管二者都由多个音频声道构成。请注意,规范虽然粗略地规定了各声道“彼此具有公认的关系”,但并未定义在声道级别访问或操作媒体的方式。实际上,根据webRTC文档中的定义,轨道的内容“将一起进行编码,以便作为某种有效负载类型(例如RTP)进行传输”。换言之,在使用对等连接进行传输时,轨道的各个声道将被视为一个单元,即使其在本地处于启用/禁用或静音状态,也不例外。  
 每个轨道都有一个源与之关联。我们稍后再解释这一现象,但现在要明确一点,即通过WebRTC不能直接访问或控制源。对源的一切控制都通过轨道实施。目前,MediaStreamTrack及其源之间的关系正在发生一些变化,但变化的方向如下:轨道不仅可以是来自源的原始媒体,还可能是浏览器提供的经过转换的版本。例如,轨道可以代表由摄像头以较高本机分辨率录制的视频的低采样率版本。

MediaStream是MediaStreamTrack对象的集合。有两种方式可用于创建这些Mediastream对象:一是通过从现有MediaSream中复制轨道来请求对本地媒体的访问;二是使用对等连接来接收新的流。目前,请求和访问本 媒体只有一种方式,即通过调用getUserMediaQ),不过将来可能会出现其他方法,例如从本地文件进行流式传输。通过MediaStream构造函数可将现有MediaStream对象的轨道“复制”到新的MediaSream对象中,该构造函数将现有的一个MediaStream对象或一系列MediaStreamTrack对象用作参数。还可以完全不指定参数,并能够在现有MediaSream中添加轨道(addTrack)和删除轨道(removeTrack()),但是这一功能尚未得到广泛支持。此外,如果不喜欢将MediaStream传递给构造函数,还可以使用clone)方法来复制流(及其所有轨道)。请注意,对于派生的媒体流(即基于其他MediaStream创建的媒体流)，MediaStream构造函数的数组参数元素不必全都来自现有的同MediaStream对象,而是允许进行混搭。各轨道也可以属于不同的类型,同一MediaStream对象可同时包含音频和视频。无论MediaStream对象是如何创建的,它们都有一个关键特征,即MediaStream对象中的所有轨道都将会在呈现时进行同步。但是,流中的各个轨道并未排序,任何添加重复轨道的尝试都将被忽略,而没有任何提示。由于每个轨道都有一个ID,且此ID会在通过对等连接时得到保留,因此对于通过对等连接发送的MediaStream,可通过在调用getAudioTracks()或getVideoTracks()后检查ID,甚至通过getTrackByld。)直接请求轨道来拼合其中的轨道。与轨道的ended状态类似,MediaStream也有一个布尔型ended属性,表示MediaSream是否已完成。如果MediaStream中所有轨道的状态均为ended,则将MediaStream视为已完成。

在实时通信中,信令的主要作用体现在四个方面:

1)协商媒体功能和设置。  
 2)标识和验证会话参与者的身份。  
 3)控制媒体会话、指示进度、更改会话和终止会话。  
 4)当会话双方同时尝试建立或更改会话时,实施双占用分解(Glare Resolution)。  
 接下来几节内容将详细探讨这些功能,说明在WebRTC中,第1项为何是必备功能,它如何实现标准化,第2、3和4项为何是可选功能或只是Web应用程序的一部分。  
 在使用标准信令协议(例如SIP或Jingle)发起实时通信时,信令通道将提供参与者的标识,并可以选择进行身份验证。在WebRTC中,除信令之外,还有两种渠道可用于确定身份。第一种渠道是参见Web应用程序提供的上下文。例如, WecbRTC网站的用户可能使用特定的屏幕名称进行登录。当此用户希望与另一用户建立会话时, Web应用程序就会将其屏幕名称作为标识提供给对方,除了相信该网站提供了准确的标识之外,另一端的用户别无选择。这与公共交换电话网线Public Switched Telephone Network, PSTN)中的呼叫方标识非常相似。PSTN用户只能相信其服务提供商向其电话呈现的呼叫方ID就是呼叫方本人,而没有其他方式进行独立确认。第二种渠道是查看URL中可能传递的标识,此URL可能包含随机令牌。在通过这种方式建立的WebRTC会话中,双方应该都知道该标识令牌。  
 WebRTC定义了另一种标识方法,即通过媒体通道,此方法不依赖于网站信息的可信性,媒体路径标识的概念最早由ZRTP [RFC6189]媒体路径密钥协议提出,根据这一概念呼叫方的标识与身份验证由媒体路径提供,完全不依赖于信令通道.webRTC采用DTLSSRTP [RFCS763]提供媒体路经标识,这通过DTLS握手期间使用的公钥的指纹实现。此指纹可使用标识提供程序(详见13.4节)进行身份验证。信令通道用于传输指纹和标识断言但不以任何其他形式参与标识断言的生成或验证。  
 HTTP轮询是一种简单的专有信令方案。例如,本章后面介绍的基于XHR的信令通道就属于HTTP轮询。  
 通过在JavaScript或jQuery中调用XHR,可使JavaScript应用程序针对Web服务器生成新的HTTP请求并处理HTTP响应。XHR是一种w3C标准API,当前定义为工作草案[XHR]，XHR正逐步成为一项标准。XHR JavaScipt API的各个功能组件均受浏览器支持。虽然XHR的名称中只包含HTTP请求,但除了发送XML请求之外,它还可以发送JavaScript对象表示法(JavaScript Object Notation，ISON)或明文。XHR可使浏览器生成新的HTTP或HTTPS请求,例如GET, POST,PUT等。相应的API调用将指定要使用的方法以及IP地址和端口号。针对请求的响应将被返回给JavaScript。  
 要使用XHR作为WebRTC信令通道, Web服务器需要运行相应的应用程序,用于通过另一个XHR通道接收HTTP请求并将从一个浏览器接收的信息以代理形式发送或转发给另一个浏览器,如图4.7所示。  
 为交换信令信息,每个浏览器中运行的JavaScript会定期向信令服务器发送HTTP消息,以进行轮询。浏览器发送的信令信息包含在POST方法中,服务器收到的信令信息包含在发送给POST的200 0K响应中,请注意,在此方案中,每个消息都是一个新请求。  
 WebRTC允许浏览器使用任意通信方法来传输信令信息,从烟雾信号到带字符串的Tin Can,再到更为快捷且更常用的方案,比如HTTP(XML HTTP请求), WebSocket和Google应用程序引攀通道,不一而足。本章前面已经探讨并比较了这些不同的方案。但请注意,无论在何种情况下,浏览器都是与Web服务器进行通信,而非另一端的浏览器。虽然不限制浏览器彼此直接通信,但Web模型本身假定,浏览器(Web客户端)只与服务器通信,而不直接与其他Web客户端通信。实际上,除非Web服务器提供了Web客户端的IP地址,否则这些客户端根本不知道如何相互联系。因此,在大多数情况下,云中都有一台服务器充当中继,用于将消息从一个浏览器发送至另一个浏览器。  
 一般而言,基于Web服务器的信令方案都属于以下两个类别之一:轮询方案和面向会话的方案。在以XHR为代表的轮询方案中,所有通信都由浏览器发起,浏览器从Web服务器接收消息的唯一方式就是接收服务器对其消息的响应。为了接收来自服务器的全部消息,浏览器需要定期向服务器发送消息来进行问询,以防存在等待接收的消息。在面向会话的方案(例如WebSocket)中,将在浏览器和web服务器之间建立一个虚拟连接,以便相互发送消息。

为简化应用程序编程操作，一种便捷的做法是定义一个信令通道接口，用于公开connect和send方法，并能够针对消息的接收时间指定处理程序。如果应用程序要在对等模式(“三角形”模式)中使用WebRTC，则该虚拟信令通道的作用是提供一个简单的面向连接的模型，用于向对等端发送信令信息。创建此虚拟连接时，有几个方面的问题需要注意。相互发送消息。  
 第一个问题是确定如何向Web服务器指示要建立的连接的性质,这个问题最为重要。此连接真的只是面向一个对等端,还是要建立群组“通话”？假定它是一对一连接,如何对另一方进行标识？一种合理的解决方案是让每个浏览器都在服务器中注册,并让服务器发回已注册浏览器的列表,以便用户选择要与哪个浏览器进行通信。另一种解决方案是使用另一端的浏览器知晓并提供的密钥。本章中的演示代码使用的就是后一种方案。在此特定演示中,使用给定密钥进行连接的第一个浏览器注册为第一方,并被告知进行等待。当第二个客户端使用同一密钥进行连接时,将在两个客户端之间建立信令通道。随后,如果有新的一方尝试使用同一密钥建立另一个连接,将销毁现有连接,并指示其进行等待。接下来,当另一个浏览器使用同一密钥进行连接时,将在这两者之间建立信令通道,以此类推,其状态就在waiting与connected之间变换。此方案的潜在缺点在于,双方都需要知道其密钥,并且需要由新的一方来销毁连接。尽管如此,它也有两个非常有利的特性;密钥可作为参数包含在URI中,这样只需将该URI添加为书签,即可保存该“连接”,而且当连接因为浏览器或网络故障而断开时,只需管理极少的状态,当连接断开时,双方只需重新加载其页面,即可重新开始通信。

附录C 软件使用说明书

首先，点击登录按钮下的注册，进入注册页面，如下图1所示：



图1 注册页面

输入完信息，注册成功后，登陆账号进入网站首页，如下图2所示：



图2 网站首页

在网站首页，能看到各个部分的信息，比如：网站公告、个人公告、调查问卷、各种提问等，点击个人中心，里面有用户的信息，各种操作的记录，像我的回答、我的调查问卷、我的研修室预定记录等等，个人信息页面如下图3所示：



图3 网站首页

附录D 主要源代码

咨询反馈子系统中问题的添加、通过模糊查询查找问题、删除问题的代码实现如下：  
 public String add(@RequestParam(value = "title")String title,  
 @RequestParam(value = "content")String content,  
 @RequestParam(value = "uid")int uid  
 ){  
 Question ask = new Question(title,content,uid);  
 questionService.addQuestion(ask);  
 return "question";  
 }  
  
 @RequestMapping("findAllQuestion")  
 @ResponseBody  
 public JSONObject findAllQuestion(@RequestParam(name = "num")int num){  
 List<Question> askLists = questionService.find();  
  
 JSONObject jsonObject = createJson(askLists,num);  
  
 return jsonObject;  
 }  
  
  
 @RequestMapping("findQuestionByStr")  
 @ResponseBody  
 public JSONObject findQuestionByStr(@RequestParam(name = "str")String str,  
 @RequestParam(name = "num")int num){  
 List<Question> askLists = questionService.findQuestionByStr(str);  
 JSONObject jsonObject = createJson(askLists,num);  
 return jsonObject;  
 }  
  
 @RequestMapping("findQuestionByAid")  
 public ModelAndView findQuestionByAid(@RequestParam(name = "aid")int aid,  
 @RequestParam(name = "str")String str){  
 Question question = questionService.findQuestionByAid(aid);  
 List<Answer> answerList = questionService.findAnswerByAid(aid);  
 String askName = this.uidToUserName(question.getUid());  
 List<String> nameList = new ArrayList();  
 if(answerList.isEmpty()){  
 }else {  
 for (Answer answer : answerList  
 ) {  
 nameList.add(this.uidToUserName(answer.getUid()));  
 }  
 }  
 ModelAndView mv = new ModelAndView();  
 mv.addObject("question",question);  
 mv.addObject("askName",askName);  
 mv.addObject("answerList",answerList);  
 mv.addObject("nameList",nameList);  
 mv.setViewName(str);  
 return mv;  
 }  
  
 @RequestMapping("findMyQuestion")  
 @ResponseBody  
 public JSONObject findMyQuestion(@RequestParam(name = "uid")int uid,  
 @RequestParam(name = "num")int num){  
 List<Question> askList = questionService.findMyQuestion(uid);  
  
 JSONObject jsonObject = createJson(askList,num);  
  
 return jsonObject;  
 }  
  
 @RequestMapping("delQuestion")  
 public String delAsk(@RequestParam(name = "aid")int aid,  
 @RequestParam(name = "str")String str){  
 questionService.delQuestionByAid(aid);  
 return str;  
 }  
  
 @RequestMapping("noAnswerQuestion")  
 @ResponseBody  
 public JSONObject noAnswerQuestion(){  
 List<Question> questionList = questionService.noAnswerQuestion();  
 JSONObject jsonObject = new JSONObject();  
 jsonObject.put("questionList",questionList);  
 return jsonObject;  
 }

回答的添加、查询并使用流加载加载数据、删除代码实现如下：  
 @RequestMapping("addAnswer")  
 public ModelAndView addAnswer(@RequestParam(name = "aid")int aid,  
 @RequestParam(name = "uid")int uid,  
 @RequestParam(name = "content")String content){  
 Answer answer = new Answer(aid,uid,content);  
 questionService.addAnswer(answer);  
 return this.findQuestionByAid(aid,"askDetail");  
 }  
 private String uidToUserName(int uid){  
 return questionService.uidToUserName(uid);  
 }  
 @RequestMapping("findMyAnswer")  
 @ResponseBody  
 public JSONObject findMyAnswer(@RequestParam(name = "uid")int uid,  
 @RequestParam(name = "num")int num){  
 List<Answer> answerList = questionService.findAnswerByUid(uid);  
 JSONObject jsonObject = createJson(num,answerList);  
 return jsonObject;  
 }  
 @RequestMapping("delAnswer")  
 public String delAnswer(@RequestParam(name = "rid")int rid,  
 @RequestParam(name = "str")String str){  
 questionService.delAnswer(rid);  
 return str;  
 }  
 public JSONObject createJson(List<Question> askLists,int num){  
 int page = askLists.size()/10;  
 if(askLists.size()%10 != 0) page++;  
 List<JSONObject> askList = new ArrayList();  
 for (int i = 10\*(num-1);i < askLists.size() && i < 10\*num; i++) {  
 JSONObject object =new JSONObject();  
 object.put("aid",askLists.get(i).getAid());  
 object.put("title",askLists.get(i).getTitle());  
 object.put("time",askLists.get(i).getTime().toString());  
 askList.add(object);  
 }  
 JSONObject jsonObject = new JSONObject();  
 jsonObject.put("count", askLists.size());  
 jsonObject.put("data", askList);  
 jsonObject.put("page",page);//总共几页  
 jsonObject.put("num",num);//当前页  
 jsonObject.put("code", 0);  
 jsonObject.put("msg", "");  
 return jsonObject;  
 }  
 public JSONObject createJson(int num,List<Answer> answerLists){  
 int page = answerLists.size()/10;  
 if(answerLists.size()%10 != 0) page++;  
 List<JSONObject> askList = new ArrayList<>();  
 for (int i = 10\*(num-1); i < answerLists.size() && i < 10\*num; i++) {  
 JSONObject object =new JSONObject();  
 Question question = questionService.findQuestionByAid(answerLists.get(i).getAid());  
 object.put("title", question.getTitle());  
 object.put("aid", question.getAid());  
 object.put("rid", answerLists.get(i).getRid());  
 object.put("time", answerLists.get(i).getAnswerTime().toString());  
 askList.add(object);  
 }  
 JSONObject jsonObject = new JSONObject();  
 jsonObject.put("count", answerLists.size());  
 jsonObject.put("data", askList);  
 jsonObject.put("page",page);//总共几页  
 jsonObject.put("num",num);//当前页  
 jsonObject.put("code", 0);  
 jsonObject.put("msg", "");  
 return jsonObject;  
 }