大连东软信息学院

**毕业设计（论文）**

**论文题目**：**在线游戏资讯平台的设计与实现**

|  |  |
| --- | --- |
| **系 所：** | 软件工程系 |
| **专 业：** | 软件工程 |
| **学生姓名：** | 王瑞鑫 |
| **学生学号：** | 16180600213 |
| **指导教师：** | 彭志豪 |
| **导师职称：** | 副教授 |
| **完成日期：** | 2020年3月17日 |

大连东软信息学院

**Dalian Neusoft University of Information**

在线游戏资讯平台的设计与实现

# 摘 要

随着现代科技的迅猛发展与人们日益生活水平的提高，娱乐的方式渐渐地向电子游戏方向转换，如今电子竞技这一娱乐以及奥运会比赛项目逐渐走进了人们的生活，各类游戏都在向自己的方向快速发展，在线游戏咨询平台可以让用户更好的查询自己想要的答案以及交流自己的游戏经验。

本次设计使用SpringBoot作为后端开发框架，Spring是一个基于java开发的功能性强，运行稳定，类库资源又极为丰富的web后端框架，SpringBoot设计简化了Spring中繁琐的配置过程，又通过内部集成了tomcat实现可以直接封装jar包运行，减轻了服务器环境搭建和管理的压力，更易于维护，且更易于集群化的部署方案。后端部分使用IDEA作为集成开发编译器，IDEA使用简单且高效，更智能的补全代码能够大大缩短开发时间成本。前端部分使用Vue框架，Vue具有组件化、数据驱动的特点，Vue从设计上讲使用的是MVVM的模式，通过数据驱动页面变化，更适合规模大、展示内容多的网站，由于组件化分散了html的代码，也使后期修改维护变得更加简单。

围绕使讨论的话题更为针对游戏这一主题，对原先各类咨询平台进行优化以及过滤，能让玩家更好的交流。可进行分类查找，玩家可对每一类游戏的帖子进行精准分类，方便其他玩家共同交流，解放了管理员需要参与管理，交给用户来完成各类上传以及删除，提高了网站的自由性。

**关键词**：游戏资讯，交流，SpringBoot，Vue

**The Design and Implementation of an Online Game Information Platform**

**Abstract**

With the rapid development of modern technology and the improvement of people's living standards, the way of entertainment is gradually transfer to playing video games. Nowadays, the entertainment of e-sports and the Olympic Games are gradually entering to people's lives. With this direction developing rapidly, the online game consulting platform allows users to better query the answers they want and exchange their gaming experience.

This design uses Spring Boot as the back-end development framework. Spring is a web-based back-end framework with strong functionality, stable operation, and rich library resources developed based on java. Spring Boot design simplifies the cumbersome configuration process in Spring. The integrated tomcat internal implementation can directly encapsulate the jar package to run whilereducing the pressure on the server environment,making it easier to build,manage and maintain, and easier to cluster deployment solutions. The back-end part uses IDEA as an integrated development compiler. IDEA is simple and efficient to use, and more intelligent completion of code can greatly reduce development time costs. The front-end part uses the Vue framework. Vue has the characteristics of componentization and data driving. Vue uses the MVVM model in design. Through data-driven page changes, it is more suitable for large-scale and large-scale websites. Due to the decentralization of components The HTML code also makes it easier to modify and maintain later.

Focusing on the topic discussion more specific to the theme of games, and to optimize and filter the original various consulting platforms to allow players to communicate better. Topics can be classified and searched, players can accurately classify the posts of each type of game, which is convenient for other players to communicate together, liberating the need for administrators to participate in management, and handing over to users to complete various uploads and deletions, improving the freedom of the website .

**Key words:** Game information, Communicate, SpringBoot, Vue

目 录

摘 要 I

Abstract I

第1章　项目概述 4

1.1 问题描述 4

1.2 项目目标 4

1.3 项目适用范围 4

1.4 涉众 4

1.5 论文的组织 4

第2章　需求分析 5

2.1 业务需求 5

2.1.1 业务流程 5

2.1.2 业务对象 6

2.2 功能性需求 6

2.2.1 用例概述 6

2.2.2 用例描述 7

2.3 非功能性需求 12

2.4 运行环境 13

第3章　系统设计 14

3.1 技术路线 14

3.2 软件体系结构设计 15

3.2.1 功能结构 15

3.2.2 系统架构 15

3.3 数据库设计 16

3.3.1 概念结构设计 16

3.3.2 表的详细设计 17

3.4 对象设计 19

3.4.1 帖子管理用例对象设计 19

3.4.2 帖子分类管理用例对象设计 21

第4章　系统实现 22

4.1 核心功能实现 25

4.1.1添加帖子功能实现 25

4.1.2添加评论功能实现 26

4.1.3管理帖子功能实现 27

4.1.4评论管理功能实现 28

第5章　系统测试 29

5.1 功能测试 29

5.2 非功能测试 30

5.3 测试总结 31

第6章　结论与展望 32

参考文献 33

致 谢 34

第1章　项目概述

**1.1 问题描述**

随着现代科技的迅猛发展与人们日益生活水平的提高，娱乐的方式渐渐地向电子游戏方向转换，如今电子竞技这一娱乐以及奥运会比赛项目逐渐走进了人们的生活，各类游戏都在向自己的方向快速发展，在线游戏咨询平台可以让用户更好的查询自己想要的答案以及交流自己的游戏经验。

使讨论的话题更为针对游戏这一主题，对原先各类咨询平台进行优化以及过滤，能让玩家更好的交流。可进行分类查找，玩家可对每一类游戏的帖子进行精准分类，方便其他玩家共同交流，解放了管理员需要参与管理，交给用户来完成各类上传以及删除，提高了网站的自由性。

**1.2 项目目标**

在网上查找攻略以及经验技巧已经成为现在每个玩家经常做的事情。平台利用高自由性可以让玩家自己在网站上查找以及交流回复，满足了一些玩家每人交流经验的问题，可以使他们找到与自己共同体验的其他玩家。降低了我们在查找攻略上耗费的多余时间。

**1.3 项目适用范围**

在用户希望找到游戏的资讯已经想与其他玩家交流时可以提供一个平台以便玩家之间的沟通与反馈。

**1.4 涉众**

平台设计人员：构思一个资讯平台系统的整体规划、模块化设计方案，在保证稳定性的情况下提高可扩展性。

系统开发人员：严格按照需求及设计文档进行功能模块开发，避免功能缺陷。

平台使用者：遵循平台的条例使用平台，用户和用户之间需要规范用语。

**1.5 论文的组织**

第一章描述了项目面向的问题目标和适用的人群。第二章描述了项目的业务需求以及对项目的基础的描述。第三章描写了项目的系统设计，对数据库和对象进行设计。第四章描写了系统实现的功能，第五章描写了对项目进行的功能测试。

第2章　需求分析

**2.1 业务需求**

**2.1.1 业务流程**

用户注册登录后进入系统，可以浏览其他用户发布的各类帖子，可以发布自己的帖子，添加帖子的游戏类型以及游戏名，玩家之间可以相互评论，管理员管理帖子和评论信息，在线游戏资讯平台的总体业务流程如图2.1所示。

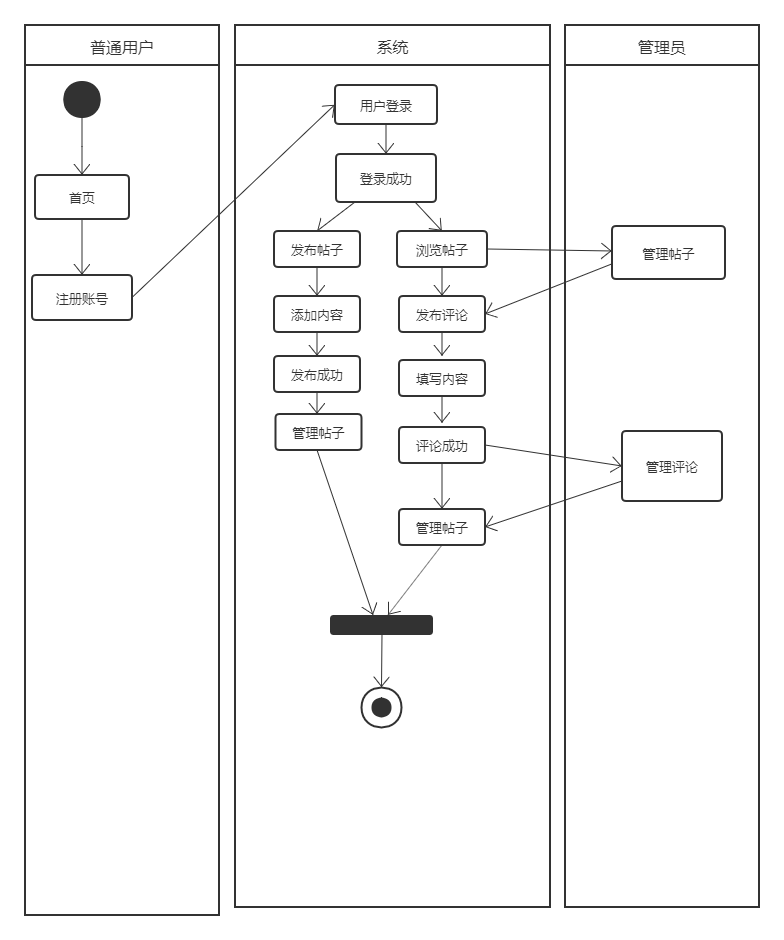


图2.1 总体业务流程图

**2.1.2 业务对象**

在线游戏资讯平台包括用户实体，评论实体，管理实体，帖子实体，类别实体，领域模型如图2.2所示。

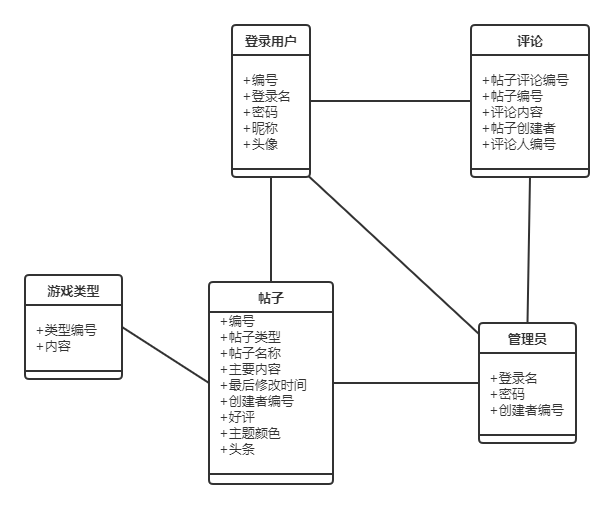


图2.2 领域模型

**2.2 功能性需求**

**2.2.1 用例概述**

在图2.3高层用例图中抽取了满足用户基本业务需求的基本用例。这些用例可以从用户，管理员完成整个在线游戏资讯平台的正常运作。

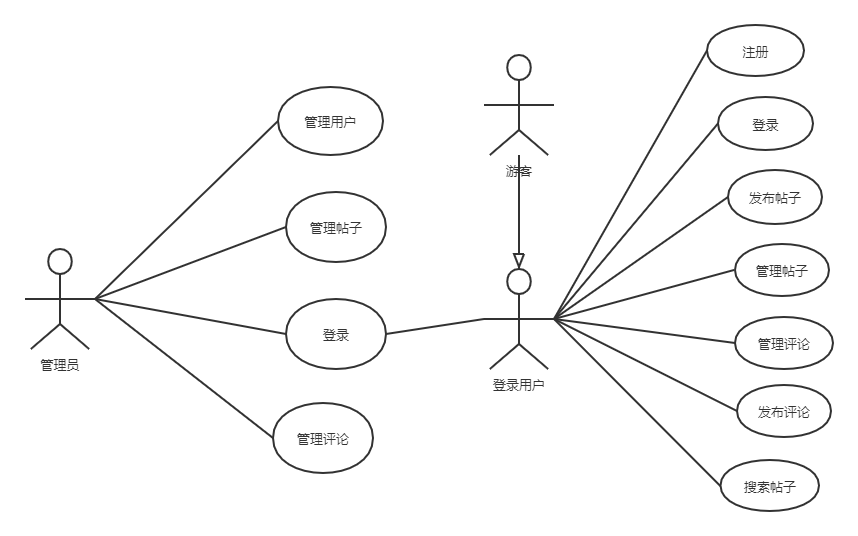


图2.3 高层用例图

用例摘要描述如表2.1所示。

表2.1 用例摘要描述

|  |  |  |
| --- | --- | --- |
| **用例标识（UC）** | **用例名称** | **摘要描述** |
| 1.1 | 注册用户 | 输入注册信息，完成用户注册操作。 |
| 1.2 | 用户登录 | 输入用户名和密码，完成用户登录操作。 |
| 1.3 | 发布帖子 | 用户登录后，发布帖子操作。 |
| 1.4 | 搜索帖子 | 输入帖子的名称，完成帖子搜索操作。 |
| 1.5 | 管理帖子 | 用户管理帖子信息，修改，删除帖子信息操作。 |
| 1.6 | 管理评论 | 用户管理评论信息，修改，删除评论信息操作。 |
| 1.7 | 发布评论 | 用户登录后，可以对帖子进行发布评论操作。 |
| 2.1 | 管理用户 | 管理员负责管理用户基本信息。 |
| 2.2 | 管理帖子 | 管理员负责管理帖子，查询帖子，删除帖子信息。 |
| 2.3 | 管理评论 | 管理评论信息，删除评论操作。 |
| 2.4 | 管理员登录 | 输入用户名和密码，完成管理员登录操作。 |

**2.2.2 用例描述**

（1）管理帖子用例

用户可以通过管理帖子用例对帖子内容进行管理操作，包括对帖子的添加，查询帖子信息，删除帖子信息。管理帖子信息用例图如图2.4所示。

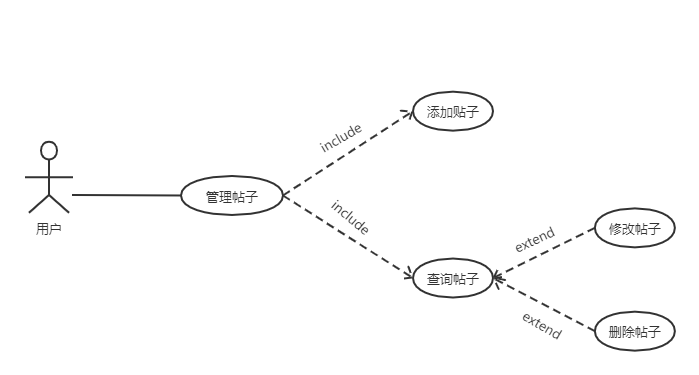


图2.4 管理帖子用例图

查询帖子信息用例描述如表2.2所示。

表2.2 查询帖子用例

|  |  |
| --- | --- |
| **用例标识** | 1.5.1 |
| **用例名称** | 用户查询帖子信息 |
| **参与者** | 登录用户 |
| **前置条件** | 用户已经登录。 |
| **后置条件** | 完成查询帖子操作。 |
| **用例概述** | 用户对帖子进行浏览查看帖子详细的操作。 |
| **基本事件流** | 1. 管理员在管理主界面选取管理帖子项。  2. 系统显示帖子列表界面。  3. 用户输入帖子信息，点击帖子详细操作。  4. 系统检查输入信息的有效性。  5. 显示查询的帖子的详细信息。 |
| **备选事件流** | 无。 |

删除帖子信息用例描述如表2.3所示。

表2.3删除帖子用例

|  |  |
| --- | --- |
| **用例标识** | 1.5.2 |
| **用例名称** | 用户删除帖子信息 |
| **参与者** | 登录用户 |
| **前置条件** | 用户已经登录。 |
| **后置条件** | 在帖子表中删除一条帖子信息。 |
| **用例概述** | 用户可以对帖子信息进行删除操作。 |
| **基本事件流** | 1. 用户在管理帖子页面选取删除帖子项。  2. 系统显示帖子列表界面，选择一条记录，点击删除按钮。  3. 用户点击确定，提交删除请求。  4. 系统检查操作的有效性。  5. 系统显示删除成功。  6. 将删除帖子信息更新到帖子信息表中 |
| **备选事件流** | 4a弹出窗口，如果确认删除，操作开始，否则操作取消，转入步骤6。 |

添加帖子用例描述如表2.4所示。

表2.4添加帖子用例

|  |  |
| --- | --- |
| **用例标识** | 1.5.3 |
| **用例名称** | 用户添加帖子信息 |
| **参与者** | 登录用户 |
| **前置条件** | 用户已经登录。 |
| **后置条件** | 在帖子信息表中添加一条帖子信息。 |
| **用例概述** | 用户可以对帖子信息进行添加操作。 |
| **基本事件流** | 1. 用户在管理主界面管理帖子信息。  2. 系统显示帖子列表界面，添加帖子操作。  3. 用户点击确定，提交添加请求。  4. 操作添加按钮，提交操作成功。  5. 将帖子信息添加到帖子信息表中 |
| **备选事件流** | 无。 |

修改帖子信息用例描述如表2.5所示。

表2.5修改帖子用例

|  |  |
| --- | --- |
| **用例标识** | 1.5.4 |
| **用例名称** | 用户修改帖子信息 |
| **参与者** | 登录用户 |
| **前置条件** | 用户已经登录。 |
| **后置条件** | 在帖子表中修改一条帖子信息。 |
| **用例概述** | 用户可以对帖子信息进行修改操作。 |

|  |  |
| --- | --- |
| **基本事件流** | 1. 用户在管理帖子页面选取修改帖子项。  2. 系统显示帖子列表界面，选择一条记录，点击修改按钮。  3. 用户点击确定，提交修改请求。  4. 系统检查操作的有效性。  5. 系统显示修改成功。  6. 将修改信息更新到帖子信息表中 |
| **备选事件流** | 4a弹出窗口，如果确认删除，操作开始，否则操作取消，转入步骤6。 |

（2）管理帖子分类用例

管理员可以通过管理帖子分类用例对帖子分类进行管理操作。管理帖子分类用例图如图2.5所示。

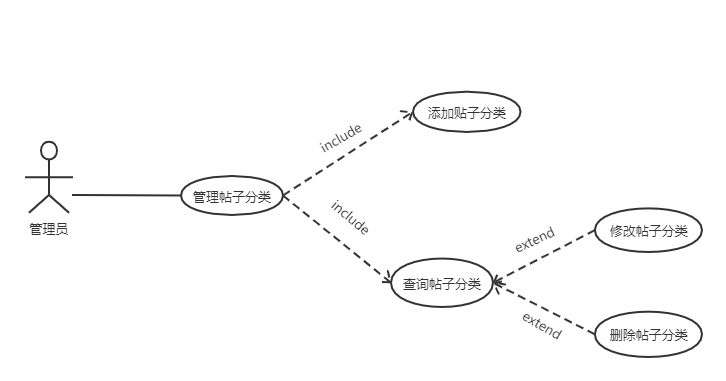


图2.5 管理帖子分类用例图

删除帖子分类信息用例描述如表2.6所示。

表2.6删除帖子分类用例

|  |  |
| --- | --- |
| **用例标识** | 2.3.1 |
| **用例名称** | 管理员删除帖子分类信息 |
| **参与者** | 管理员 |
| **前置条件** | 已经以管理员身份登录。 |
| **后置条件** | 在帖子分类信息表中删除一条帖子分类信息。 |
| **用例概述** | 管理员可以对帖子分类信息进行删除操作。 |
| **基本事件流** | 1. 管理员在管理主界面选取管理帖子分类项。  2. 系统显示帖子分类列表界面，选择一条记录，点击删除按钮。  3. 管理员点击确定，提交删除请求。  4. 系统检查操作的有效性。  5. 系统显示删除成功。  6. 将删除帖子分类信息更新到帖子分类信息表中 |
| **备选事件流** | 4a弹出窗口，如果确认删除，操作开始，否则操作取消，转入步骤6。 |

添加帖子分类用例描述如表2.7所示。

表2.7添加帖子分类用例

|  |  |
| --- | --- |
| **用例标识** | 2.3.2 |
| **用例名称** | 管理员添加帖子分类信息 |
| **参与者** | 管理员 |
| **前置条件** | 已经以管理员身份登录。 |
| **后置条件** | 在帖子分类表中添加帖子分类信息。 |
| **用例概述** | 管理员可以对帖子分类信息进行添加操作。 |
| **基本事件流** | 1.在主界面选取管理景点分类项。  2. 系统显示帖子分类列表界面，输入帖子分类，添加操作。  3. 管理提出添加帖子分类的数据库请求。  4. 系统检查操作的有效性，接受添加操作请求。  5. 操作完成，提交操作成功。  6. 将添加信息更新到帖子分类信息表中 |

续表2.7添加帖子分类用例

|  |  |
| --- | --- |
| **备选事件流** | 无。 |

查询帖子分类信息用例描述如表2.8所示。

表2.8 查询帖子分类用例

|  |  |
| --- | --- |
| **用例标识** | 2.3.3 |
| **用例名称** | 管理员查询帖子分类 |
| **参与者** | 管理员 |
| **前置条件** | 已经以管理员身份登录。 |
| **后置条件** | 完成查询帖子分类操作。 |
| **用例概述** | 管理员对帖子分类进行浏览操作。 |
| **基本事件流** | 1. 管理员在管理主界面选取管理帖子分类项。  2. 系统显示用户列表界面。  3. 管理员点击某一条帖子分类信息操作。  4. 系统检查输入信息的有效性。  5. 显示查询出来的帖子分类信息。 |
| **备选事件流** | 无。 |

修改帖子分类信息用例描述如表2.9所示。

表2.9修改帖子分类用例

|  |  |
| --- | --- |
| **用例标识** | 2.3.4 |
| **用例名称** | 管理员修改帖子分类信息 |
| **参与者** | 管理员 |
| **前置条件** | 已经以管理员身份登录。 |
| **后置条件** | 在帖子分类表中修改一条帖子分类信息。 |
| **用例概述** | 管理员可以对帖子分类信息进行修改操作。 |
| **基本事件流** | 1. 管理员在管理帖子分类页面选取管理帖子分类项。  2. 系统显示帖子分类列表界面，选择一条记录，点击修改按钮。  3. 管理员点击确定，提交修改请求。  4. 系统检查操作的有效性。  5. 系统显示修改成功。  6. 将修改信息更新到帖子分类信息表中 |
| **备选事件流** | 4a弹出窗口，如果确认删除，操作开始，否则操作取消，转入步骤6。 |

（3）管理评论用例

用户可以通过管理评论用例对评论信息进行管理操作，包括浏览评论信息，删除评论信息。管理评论信息用例图如图2.6所示。

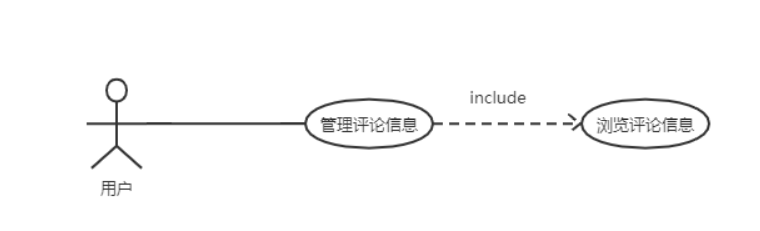


图2.6 管理评论用例图

浏览评论信息用例描述如表2.10所示。

表2.10 浏览评论用例

|  |  |
| --- | --- |
| **用例标识** | 2.1.1 |
| **用例名称** | 用户浏览评论用例 |
| **参与者** | 用户 |
| **前置条件** | 用户已经登录。 |
| **后置条件** | 完成浏览评论操作。 |
| **用例概述** | 用户对评论进行浏览操作。 |
| **基本事件流** | 1. 用户在管理主界面选取管理评论项。  2. 系统显示评论列表界面。  3. 用户点击某一条评论信息操作。  4. 系统检查输入信息的有效性。  5. 显示查询出来的评论信息。 |
| **备选事件流** | 无。 |

**2.3 非功能性需求**

本系统能在Windows7/8/10操作系统平台环境下正常运行，而且目前在同一个局域网下，可以共同使用，需要MySQL数据库与Tomcat服务器的支持，当用户的需求发生变化时，本系统的操作方法、运行环境基本不会发生变化，被改变的只是数据库文件中的记录。

1、系统的安全性

网站采用二级安全保障。第一级：依赖于互联网自身对系统登陆用户的使用权限的界定。第二级：在系统中利用管理员的权限去操作用户信息对用户的登陆权限做以界定。

2、系统的可靠性

网站数据的可利用性对用户管理工作来说是尤为重要的，本系统要求使用功能比较优秀的数据库系统。

3、系统的可维护性

网站的日常系统维护工作量非常大，因为是面对所有群体， 所以要求该系统的设计理念能够让该网站的维护人员很容易理解。

4、系统的健壮性

网站能够检测自己内部的设计或者编码错误，并得到正确的执行结果，并且可以自行判断出正确的输入。

5、系统的易使用性

网站使用户能理解本网站是否合适以及如何能将本网站用于特定的任务和使用条件的能力。

6、系统的可扩充性

本网站可以更进一步的加强管理和增添新的功能以满足用户的需求，拥有较强的可扩充性。

7、系统的可重用性

本网站开发的全生命周期都有可重用的价值，包括项目的组织、软件需求、设计、文档、实现、测试方法和[测试用例](https://www.baidu.com/s?wd=%E6%B5%8B%E8%AF%95%E7%94%A8%E4%BE%8B&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)都是可以被重复利用或借鉴的有效资源。

**2.4 运行环境**

本网站运行所需要的硬件环境如下。

1、软件环境

客户端：Windows 7/8/10 IE8.0以上版本。

服务器端：Windows 7/8/10 IE8.0以上版本。

2、硬件环境

服务器： CPU的主频2.0GH以上或AMD2500+以上，内存1G以上。

客户机：CPU要求2.0GH以上，内存1G以上。

3、支持软件

操作系统：系统可在Windows XP/7/8以上操作系统平台环境下正常运行。

软件环境：操作系统 + Tomcat6.0以上 +MYSQL

第3章　系统设计

**3.1 技术路线**

在线游戏资讯平台是一个基于Spring Boot搭建的web项目。前端用到了JavaScript，HTML,CSS，VUE等技术，数据库使用的是MySQL。

MVC全名是Model View Controller，是模型(model)－视图(view)－控制器(controller)的缩写，一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。MVC被独特的发展起来用于映射传统的输入、处理和输出功能在一个逻辑的图形化用户界面的结构中。

本次设计使用SpringBoot作为后端开发框架，Spring是一个基于java开发的功能性强，运行稳定，类库资源又极为丰富的web后端框架，SpringBoot设计简化了Spring中繁琐的配置过程，又通过内部集成了tomcat实现可以直接封装jar包运行，减轻了服务器环境搭建和管理的压力，更易于维护，且更易于集群化的部署方案。后端部分使用IDEA作为集成开发编译器，IDEA使用简单且高效，更智能的补全代码能够大大缩短开发时间成本。前端部分使用Vue框架，Vue具有组件化、数据驱动的特点，Vue从设计上讲使用的是MVVM的模式，通过数据驱动页面变化，更适合规模大、展示内容多的网站，由于组件化分散了html的代码，也使后期修改维护变得更加简单。前端开发环境使用Visual Studio Code作为开发工具，Node.js作为开发环境，Visual Studio Code轻量易用，通过集成插件的方式扩展对语法的支持，非常适合前端的开发，Nodejs可以通过webpack、node-sass等依赖扩展库辅助Vue开发，并将开发结果轻量化打包成直接可以由浏览器执行的内容。

MySQL是一种关系数据库管理系统，关系数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。

MySQL所使用的 SQL 语言是用于访问[数据库](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93" \t "_blank)的最常用标准化语言。MySQL 软件采用了双授权政策，分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其是[开放源码](https://baike.baidu.com/item/%E5%BC%80%E6%94%BE%E6%BA%90%E7%A0%81" \t "_blank)这一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库。

**3.2 软件体系结构设计**

**3.2.1 功能结构**

根据需求调研结果确定本系统主要包括以下功能模块，如图3.1所示。

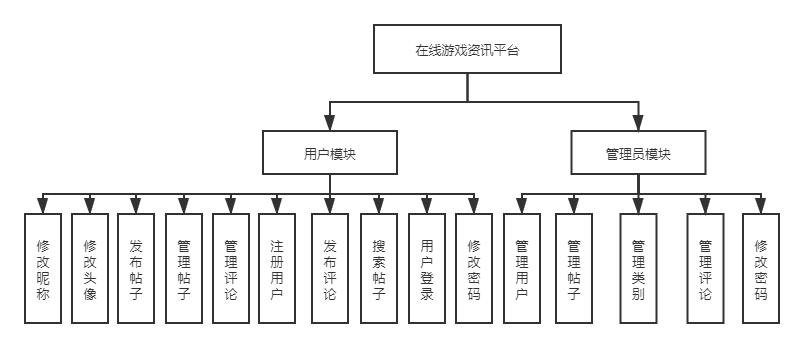


图3.1系统功能结构图

**3.2.2 系统架构**

系统架构图如图3.2所示。本网站是基于MVC框架基础上来开发，分为表现层，业务逻辑层，数据访问层。View表现层是将数据库查询出来的数据展现到用户的界面，与用户交互，完成数据的显示并且接收和校验用户的每个请求，业务逻辑层则是程序的核心，Controller类，接受并处理用户的每个请求并且向访问层发出系统的处理结果，数据访问层是接受系统的处理结果并对数据库进行更新操作。主要模块和业务的功能是在业务逻辑层的部分进行设计并实现的。

系统架构图如图3.2所示。

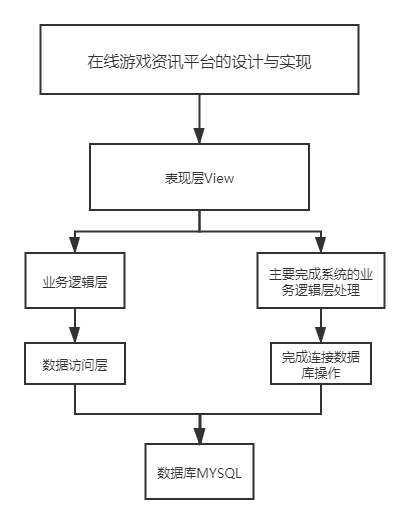


图3.2 系统架构图

**3.3 数据库设计**

**3.3.1 概念结构设计**

帖子分类和帖子是一对多的关系，帖子和评论是一对多关系，用户和评论是一对多关系，用户和帖子是一对多关系，管理员和帖子是一对多关系，管理员和帖子类别是一对多关系，管理员和评论是一对多关系，系统概念模型（E-R图）如图3.3所示。

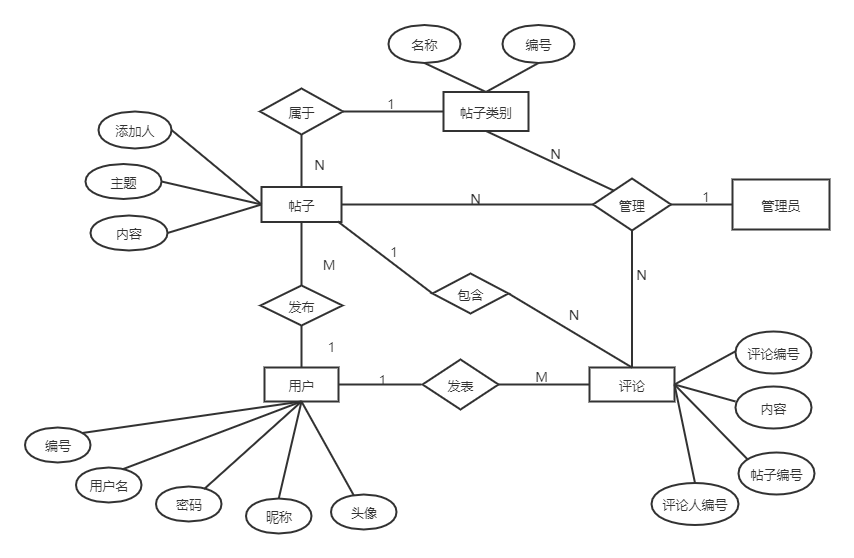


图3.3 E-R图

**3.3.2 表的详细设计**

通过对领域对象、业务对象及对象间关联的分析，对本系统的数据库表进行了设计，如表3.1-表3.5所示。本系统数据库管理采用MySQL。

（1）管理员表(admin)

管理员表主要用于保存管理员的信息，主要字段包括：账号、密码、创建者。表结构如表3.1所示。

表3.1 管理员表（admin）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **数据类型** | **允许空** | **是否主键** | **说明** |
| adminname | varchar | 是 | 是 | 管理员名称 |
| adminpass | varchar | 否 | 否 | 管理员密码 |
| createby | varchar | 否 | 否 | 创建者 |

（2）游戏类型分类表(bbstag)

游戏类型分类表主要用于保存游戏类型分类的信息，主要字段包括：游戏类型编号、游戏类型名称。表结构如表3.2所示。

表3.2 游戏类型分类表（bbstag）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **数据类型** | **允许空** | **是否主键** | **说明** |
| tagId | int | 是 | 是 | 编号 |
| tagName | varchar | 否 | 否 | 内容 |

（3）游戏表(bbstype)

游戏表主要对游戏类型进行细分，主要字段包括：游戏编号、游戏名、游戏类型编号。表结构如表3.3所示。

表3.3 游戏表（bbstype）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **数据类型** | **允许空** | **是否主键** | **说明** |
| typeId | int | 否 | 是 | 游戏编号 |
| typeName | varchar | 否 | 否 | 游戏名 |
| tagId | int | 否 | 否 | 游戏类型编号 |

（4）用户信息表(user)

用户信息表主要用于保存用户信息，主要字段包括：编号、登录名、密码、昵称、头像。表结构如表3.4所示。

表3.4 用户信息表（user）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **数据类型** | **允许空** | **是否主键** | **说明** |
| userId | int | 否 | 是 | 编号 |
| username | varchar | 否 | 否 | 登录名 |
| password | varchar | 否 | 否 | 密码 |
| nickname | varchar | 否 | 否 | 昵称 |
| headimage | varchar | 否 | 否 | 头像 |

（5）评论信息表(comment)

评论信息表主要用于保存评论的信息，主要字段包括：评论编号、帖子编号、内容、创建者、评论人。表结构如表3.5所示。

表3.5评论信息表（comment）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **数据类型** | **允许空** | **是否主键** | **说明** |
| noteCommentId | int | 否 | 是 | 评论编号 |
| noteId | int | 否 | 否 | 帖子编号 |
| commentText | text | 否 | 否 | 内容 |
| createBy | int | 否 | 否 | 创建者 |
| refer\_id | int | 否 | 否 | 评论人 |

（6）帖子信息表(note)

帖子信息表主要用于保存帖子的信息，主要字段包括：帖子编号、类型编号、名称、内容、最后修改时间、创建者、好评、标题颜色、置顶。表结构如表3.6所示。

表3.6帖子信息表（note）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **列名** | **数据类型** | **长度** | **允许空** | **是否主键** | **说明** |
| noteId | int | 11 | 否 | 是 | 帖子编号 |
| typeId | int | 50 | 否 | 否 | 类型编号 |
| noteName | varchar | 30 | 否 | 否 | 名称 |
| note | text | 50 | 否 | 否 | 内容 |
| lastEditTime | datetime | 50 | 否 | 否 | 最后修改时间 |
| createBy | int | 50 | 否 | 否 | 创建者 |
| isGood | int | 100 | 否 | 否 | 好评 |
| tittleColor | varchar | 200 | 否 | 否 | 标题颜色 |
| top | int | 30 | 否 | 否 | 置顶 |



**3.4 对象设计**

本系统的功能包括帖子管理、帖子分类管理、评论管理等。下面对各个模块进行详细设计。

**3.4.1 帖子管理用例对象设计**

（1）管理帖子用例实现

管理帖子用例的实现顺序图如图3.4所示。

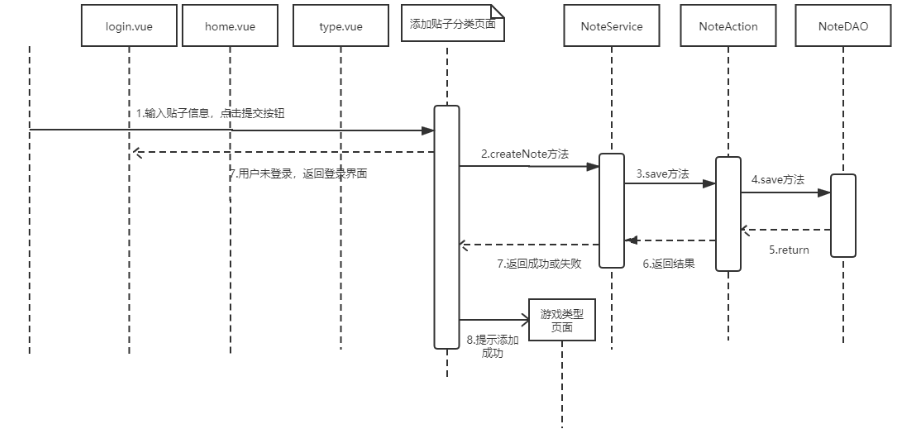


图3.4管理帖子用例顺序图

说明：用户对帖子进行管理，在添加帖子页面，输入帖子信息，点击添加按钮，调用createNote方法，NoteService向NoteDAO提交请求，NoteDAO返回结果，完成添加添加景点的操作。

（2）设计类图

根据顺序图的分析得到管理帖子用例的设计类图，如图3.5所示。

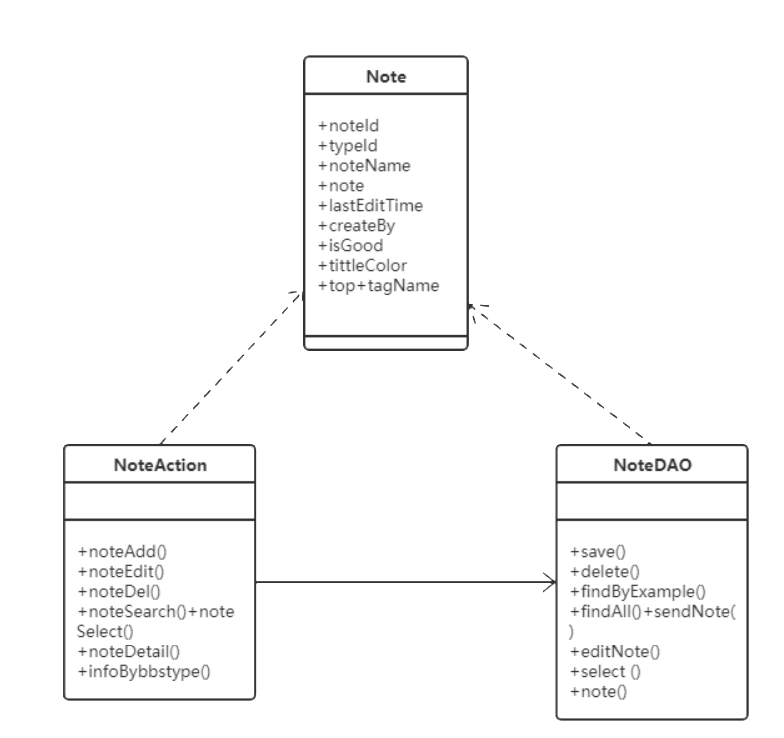


图3.5 管理帖子用例的设计类图

类图中NoteService包中类的描述如表3.7所示。

表3.7 Note包中类的描述

|  |  |  |
| --- | --- | --- |
| **类名** | **属性/方法** | **描述** |
| NoteService | +noteId | 帖子编号 |
| +typeId | 类型编号 |
| +noteName | 帖子主题 |
| +note | 帖子内容 |
| +lastEditTime | 最后修改时间 |
| +createBy | 创建者 |
| +isGood | 好评 |
| +tittleColor | 标题颜色 |
| +top | 置顶 |
| +tagName | 类型名 |

类图中NoteDAO包中类的描述如表3.8所示。

表3.8 NoteAction包中类的描述

|  |  |  |
| --- | --- | --- |
| **类名** | **属性/方法** | **描述** |
| NoteAction | +noteadd() | 添加帖子的方法 |
| +noteEdit () | 编辑帖子的方法 |
| +noteDel () | 删除帖子的方法 |
| +notesearch() | 搜索帖子的方法 |
| +noteselect() | 查询帖子的方法 |
| +notedetail() | 查看帖子的详细信息的方法 |
| +infoBybbstype() | 根据类别查询帖子分类方法 |

表3.9 NoteDAO包中类的描述

|  |  |  |
| --- | --- | --- |
| **类名** | **属性/方法** | **描述** |
| NoteDAO | +save() | 保存帖子的方法 |
| +delete() | 删除帖子的方法 |
| +findByExample() | 根据类型查询帖子的方法 |
| +findAll() | 查询帖子所有记录的方法 |

**3.4.2 帖子分类管理用例对象设计**

（1）管理帖子分类用例实现

管理帖子分类用例的实现顺序图如图3.6所示。

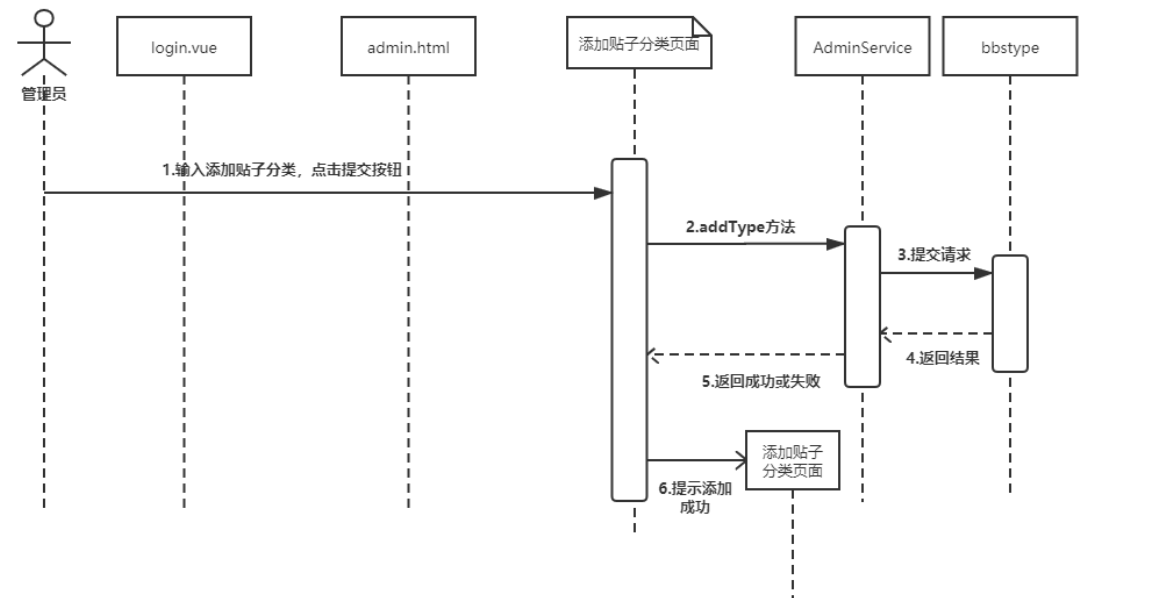


图3.6管理帖子分类用例顺序图

说明：管理员对帖子分类进行管理，在帖子分类列表页面，输入添加帖子分类，调用addType方法，AdminService向bbstype发送请求，bbstype类里面存在处理添加类别信息的方法操作。

（2）设计类图

根据顺序图的分析得到管理景点分类用例的设计类图，如图3.7所示。

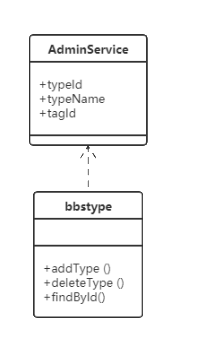


图3.7 管理类别用例的设计类图

类图中AdminService包中类的描述如表3.10所示。

表3.10 TFenlei包中类的描述

|  |  |  |
| --- | --- | --- |
| **类名** | **属性/方法** | **描述** |
| AdminService | +typeId | 分类编号 |
| +typeName | 分类名称 |
| +tagId | 类型编号 |

类图中bbstype包中类的描述如表3.11所示。

表3.11 bbstype包中类的描述

|  |  |  |
| --- | --- | --- |
| **类名** | **属性/方法** | **描述** |
| bbstype | +addType () | 添加帖子分类的方法 |
| +deleteType () | 删除帖子分类的方法 |
| +findById() | 更改帖子分类的方法 |

**3.4.3 评论管理用例对象设计**

（1）管理评论用例实现

管理帖子分类用例的实现顺序图如图3.8所示。

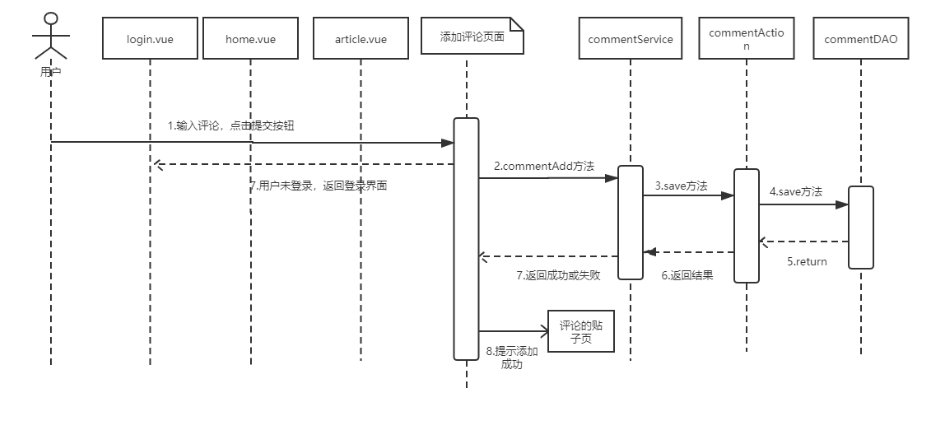


图3.8管理评论用例顺序图

说明：用户对评论进行管理，在添加评论页面，输入评论，点击确定按钮，调用commentAction类里面的commentAdd方法，commentAction调用commentDAO类里面的save方法，commentDAO类里面save的方法，完成添加评论的操作。

（2）设计类图

根据顺序图的分析得到管理景点分类用例的设计类图，如图3.9所示。

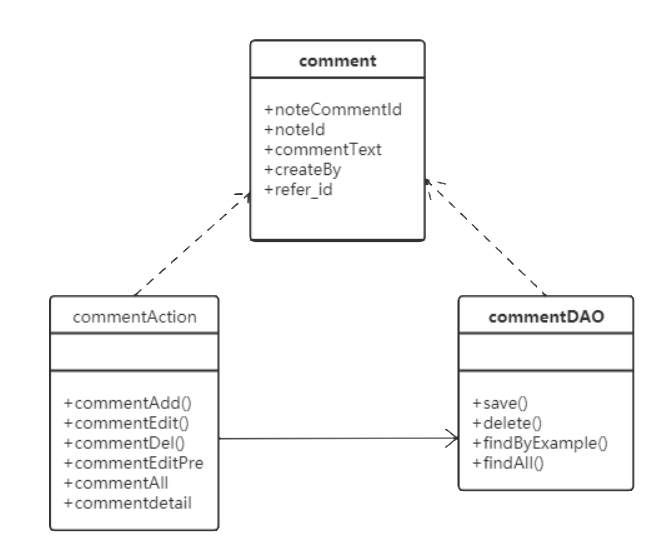


图3.9 管理类别用例的设计类图

类图中comment包中类的描述如表3.12所示。

表3.12comment包中类的描述

|  |  |  |
| --- | --- | --- |
| **类名** | **属性/方法** | **描述** |
| NoteService | +noteCommentId | 帖子中评论编号 |
| +noteId | 帖子编号 |
| +commentText | 评论内容 |
| +createBy | 评论添加者 |
| +refer\_id | 评论者编号 |

类图中commentAction包中类的描述如表3.13所示。

表3.13 commentAction包中类的描述

|  |  |  |
| --- | --- | --- |
| **类名** | **属性/方法** | **描述** |
| NoteAction | +commentAdd() | 添加评论的方法 |
| +commentEdit() | 编辑评论的方法 |
| +commentDel() | 删除评论的方法 |
| +commentEditPre | 加载评论的方法 |
| +commentAll | 加载所有评论的方法 |
| +commentdetail | 查看帖子评论的详细信息的方法 |

表3.14 commentDAO包中类的描述

|  |  |  |
| --- | --- | --- |
| **类名** | **属性/方法** | **描述** |
| NoteDAO | +save() | 保存评论的方法 |
| +delete() | 删除评论的方法 |
| +findByExample() | 根据类型查询评论的方法 |
| +findAll() | 查询帖子中评论所有记录的方法 |

第4章　系统实现

**4.1 核心功能实现**

**4.1.1添加帖子功能实现**

在添加帖子界面选择游戏分类，输入标题，内容，点击提交按钮，完成添加帖子操作，页面如图4.1所示：

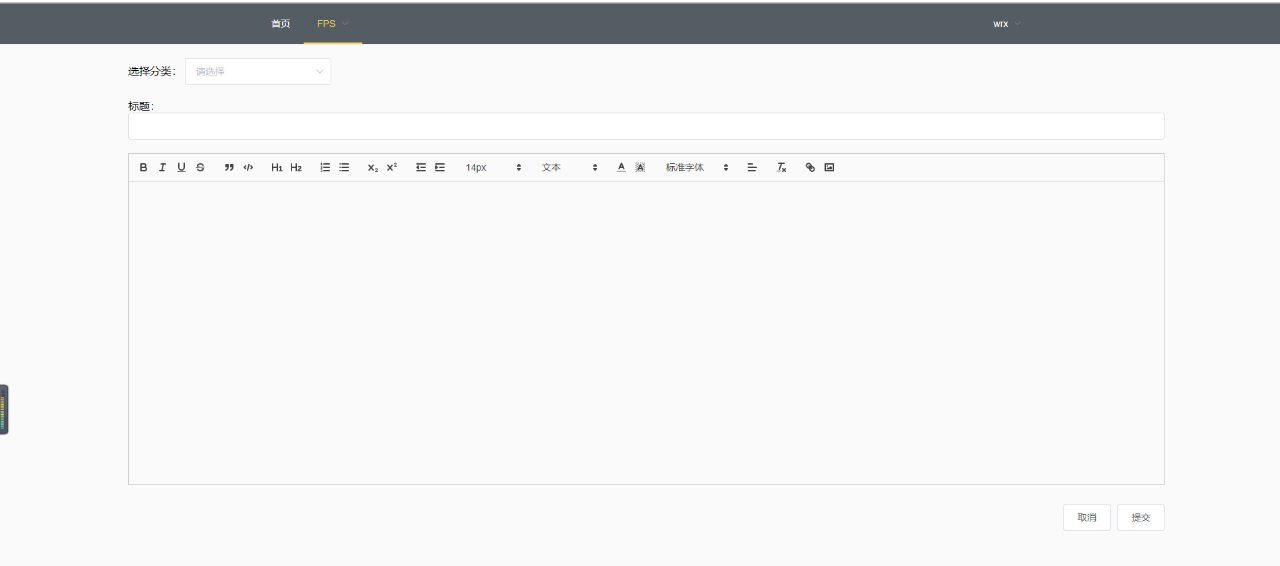


图4.1 添加公告页面

主要代码：

<template>

  <div class="page-continue">

    <div class="el-line">

      选择分类：

      <el-cascader :options="tags" v-model="typeSelect"></el-cascader>

    </div>

    <div class="el-line">

      标题：

      <el-input v-model="title"></el-input>

    </div>

    <div class="el-line">

      <Editor class="editor" v-model="text"></Editor>

    </div>

    <div class="el-line button-line">

      <el-button @click="back">取消</el-button>

      <el-button @click="update">提交</el-button>

    </div>

  </div>

</template>

**4.1.2添加评论功能实现**

在帖子界面点击评论，写下评论内容，点击确定按钮，完成添加评论操作，页面如图4.2所示：

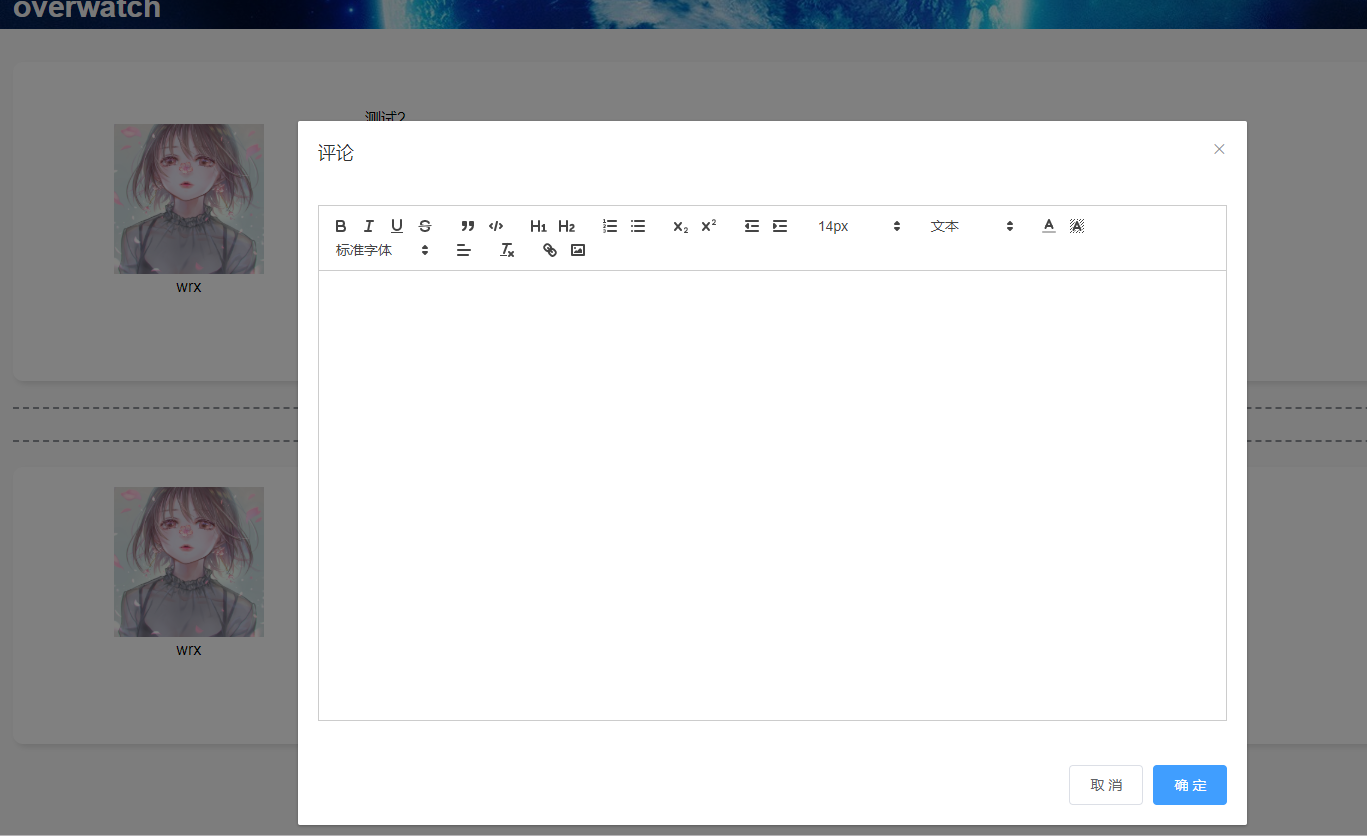


图4.2 发布景点页面

主要代码：

<div class="panel article">

        <div class="inline">

          <div class="user">

            <img :src="author.headimg">

            <div class>{{author.nickname}}</div>

          </div>

          <div class="content" v-html="text"></div>

        </div>

        <div class="buttonline">

          <el-button @click="comment" type="text">评论</el-button>

        </div>

      </div>

      <div class="line">评论</div>

 <el-dialog title="评论" :visible.sync="commentouterVisible">

      <Editor class="editor" v-model="commentText"></Editor>

      <div slot="footer" class="dialog-footer">

        <el-button @click="commentouterVisible = false">取 消</el-button>

        <el-button type="primary" @click="commentSubmit">确 定</el-button>

      </div>

    </el-dialog>

**4.1.3管理帖子功能实现**

管理帖子，点击删除按钮操作，完成删除景点信息，页面如图4.3所示：



图4.3 删除景点页面

主要代码：

@PostMapping("/note/delete")  
@ApiOperation("管理员删除帖子")  
public HttpSendBasic admindeleteNote(@RequestParam("noteId") Integer noteId) {  
 return adminService.deleteNote(noteId);  
}

4.1.4添加帖子分类实现

发布帖子分类的基本信息操作，输入帖子分类内容，完成帖子分类操作如图4.4所示：



图4.4 景点分类页面

主要代码：

@PostMapping("/type/add")  
@ApiOperation("管理员添加分类")  
public HttpSendBasic adminAddType(@RequestParam("tagId") Integer tagId,@RequestParam("typeName") String typeName,  
@RequestParam(name = "icon",required = false) String icon){  
 return adminService.addType(tagId, typeName, icon);  
}

**4.1.4评论管理功能实现**

管理评论信息，点击删除按钮操作，完成删除评论信息，页面如图4.5所示：



图4.5 评论管理页面

主要代码：

@PostMapping("/comment/delete")  
@ApiOperation("管理员删除评论")  
public HttpSendBasic adminDeleteComment(@RequestParam("commentId") Integer commentId){  
 return adminService.deleteComment(commentId);  
}

第5章　系统测试

游戏资讯平台的测试本着及早接入的原则，因此早在需求阶段测试人员就对需求的覆盖度、数据模型的正确性都进行了相应的验证测试，在系统开发过程中进行了单元测试，在集成阶段进行了集成测试，在整个系统开发完成后又针对功能性方面进行了系统测试。由于篇幅的限制，下面主要围绕功能测试用例、性能测试和测试总结三个方面进行说明。

**5.1 功能测试**

本系统主要对查找以及编写上传进行测试：

（1）查找测试

查找测试包括查找帖子。测试用例如表5.1所示。

表5.1查找帖子功能测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **用例**  **编号** | **测试用例描述** | **操作过程及数据** | **预期结果** | **测试结果** |
| findpost\_01 | 正确填写查找信息 | 按照查找的要求在查找框中铁屑查找信息 | 系统跳转查找的结果 | 系统正确跳转到查找结果 |
| findpost \_02 | 类型的选择不正确 | 查找的帖子不在指定的游戏类型中 | 系统提示游戏的类型查找不到该帖子 | 找不到查找的贴子 |
| findpost \_03 | 查找的帖子不存在 | 正常查找帖子但是不存在 | 系统提示该帖子不存在或已删除 | 查找不到不存在的贴子，并提示 |

（2）编写上传测试

编写上传测试包括编写帖子，对别人的帖子进行评论。测试用例如表5.2-5.3所示。

表5.2 编写帖子功能测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **用例**  **编号** | **测试用例描述** | **操作过程及数据** | **预期结果** | **测试结果** |
| addpost\_01 | 正确编写添加帖子的内容 | 正确编写添加帖子的游戏种类，游戏名以及帖子内容 | 系统提示添加成功 | 系统提示添加成功 |
| addpost\_02 | 未选择游戏的种类 | 只填写内容，没有选择游戏的种类 | 系统提示没有选择游戏种类 | 正确提示没有选择游戏种类 |
| addpost \_03 | 未填写游戏名 | 选择了游戏种类但没有写游戏名 | 系统提示填写正确的游戏名 | 提示没有填写正确的游戏名 |
| addpost \_04 | 未填写主要内容 | 主要内容输入为0 | 系统提示没有输入主要内容 | 提示没有填写主要内容 |

表5.3 对别人的帖子进行评论功能测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | **测试用例描述** | **操作过程及数据** | **预期结果** |
| comment \_01 | 正确评论帖子 | 正确评论帖子进行发表 | 系统提示评论成功 |
| comment \_02 | 没有填写评论 | 评论总输入字符为0 | 系统提示没有输入评论内容 |

（3）添加帖子模块功能测试

添加帖子模块功能测试包括添加帖子信息。测试用例设计如表5.3所示。

表5.4 添加帖子模块功能测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **用例编号** | **测试用例描述** | **操作过程及数据** | **预期结果** | **测试结果** |
| D006 | 添加帖子信息 | 输入帖子基本信息，点击添加按钮 | 添加帖子成功 | 通过 |

（4）添加帖子分类功能测试

添加帖子分类功能测试包括添加帖子信息。测试用例设计如表5.4所示。

表5.5 添加帖子分类功能测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **用例编号** | **测试用例描述** | **操作过程及数据** | **预期结果** | **测试结果** |
| D007 | 添加帖子分类信息 | 输入帖子分类信息，点击添加按钮 | 添加帖子分类成功 | 通过 |

（5）发表评论功能测试

发表评论功能测试包括发表评论。测试用例设计如表5.6所示。

表5.6 发表评论功能测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **用例编号** | **测试用例描述** | **操作过程及数据** | **预期结果** | **测试结果** |
| D009 | 发表评论 | 输入评论信息，点击发表按钮 | 发表评论成功 | 通过 |

（6）管理评论功能测试

管理评论功能测试包括删除评论。测试用例设计如表5.7所示。

表5.7 管理评论功能测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **用例编号** | **测试用例描述** | **操作过程及数据** | **预期结果** | **测试结果** |
| D010 | 删除评论 | 点击某条评论，点击删除按钮 | 删除评论成功 | 通过 |

**5.2 非功能测试**

表5.8 手机型号兼容性测试用例

|  |  |
| --- | --- |
| **测试用例名称** | **测试手机型号** |
| 兼容性\_001 | 小米 MIX3 |
| 兼容性\_002 | 努比亚红魔 |
| 兼容性\_003 | 小米 6 |

表5.9 浏览器兼容性测试用例

|  |  |
| --- | --- |
| **测试用例名称** | **浏览器版本** |
| 兼容性\_001 | IE 10 |
| 兼容性\_002 | Chrome 62.0.3202.94 |
| 兼容性\_003 | Firefox 57.0.0.6525 |

性能测试用例如表5.11所示。

表5.10 兼容性测试用例

|  |  |  |
| --- | --- | --- |
| **场景** | **并发数量** | **测试点** |
| 登录 | 500 | 登录的响应时间 |
| 1000 |
| 2300 |
| 2400 |
| 查找贴子 | 500 | 查找的响应时间 |
| 1000 |
| 2300 | 查找的响应时间 |
| 2400 |
| 评论 | 500 | 500人集合评论的负载情况 |
| 1000 | 1000人集合评论的负载情况 |

**5.3 测试总结**

测试用例共设计了3个，执行了3个，通过2个，在兼容性测试中存在一定的问题，在2400人同时查找贴子时，项目存在并发数量过高导致卡死。

第6章　结论与展望

本系统在设计之初充分考虑了用户与用户之间的需求，并针对现实存在的一些问题作出了相应的分析，为能够做出一个更好更便捷的系统做出准备。最后选择了最适宜开发的应用平台和编程语言进行系统的开发，系统的主要功能和模块的决定也是经过反复思考才确定的，最终才使得系统具有强大的处理功能，有利于用户更好的操作。

本系统现阶段虽然已经初步完成，但是仍然在不断更新。需要在使用过程中发现本系统的不足和缺陷，然后进一步完善。在本系统实践的过程中要进一步完善系统的安全性，易用性，用户对我们系统的好评是我们莫大的动力。考虑随着系统用户越来越多，存储的数据量越来越大，可能导致系统运行效率降低，有的同事建议用分库等解决方案，为了保证数据的严谨性，系统设计和架构可能才存在一些不足之处，这样我们多学习，才能把系统设计得更好。

经过这次毕业设计，我明白了想把一个项目做好，需要良好的表达和沟通能力，不怕麻烦，不怕失败，反复调式，细心，仔细，重要的是一定要深入了解业务需求，多站在用户的角度想想才会做出最佳的系统.

在此之后，我准备以以下的几种方式来推广我的网站。首先是内容，我们第一可以转载，第二，我们可以去国外的网站翻译相关的内容，第三，我们也可以自己原创。第四，我们还可以鼓励别人多发布帖子。第五，我们也可以伪原创别人的。其次是评论推广，我们可以经常去一些和我们主题相关的博客去评论，还有一些你目标人群的博客。往往效果会出乎意料的好。因为做博客的90%都是很在意和很好奇你对他的评论的。特别是一些流量少的博客。选择博客评论主要有几个好处，第一是可以留外链。第二，博主大部分会回访。第三，访客也会注意到你的。目前大部分的博客都是可以留外链的。通过这些方法，希望能够成功的推广我的网站。

此次系统开发，尽量确保了系统的进度方面和质量方面的把关，但因为时间上的关系，系统虽已经完成，但难免存在其他没有想到的不足之处需要好好改进和完善。

参考文献

[1] 朱运乔. 基于SpringBoot+SSM框架的Web应用系统搭建与实现.电脑编程技巧与维护[J]. 2019,(10),23-25

[2] 刘敏. 基于SpringBoot框架社交网络平台的设计与实现[J]. 社交网络服务, TP311.52

[3] 朱二华.基于Vue.js的Web前端应用研究[J]. 10.15913/j.cnki.kjycx.2017.20.119

[4] 王志任.基于Vue.js的开发平台的设计与实现[J]. F270.7;TP311.52

[5] 冯传波 彭章友 张钟浩. 基于Vue.js的移动应用可视化平台的研究[J]. 工业控制计算机 2019,32(05),102-103

[6] Olga Filipova  . Learning Vue.js 2: Learn how to build amazing and complex reactive web applications easily with Vue.js (English Edition).

[7] 张丽. 以HTML 5+CSS3+jQuery为基础的响应式布局网页设计探讨. TP393.092

[8] 黄悦深. 基于HTML5的移动Web App开发. 10.13663/j.cnki.lj.2014.07.012

[9] 范兆忠张海攀魏跃堂. HTML 5与CSS 3在网页中的应用. 无线互联科技 2015,(15),32-33

[10] 贾轩 王栋轩. 基于SSM框架下用户注册登录界面的设计与实现[J].  信息系统工程 2019,(02),54

[11] 王建翠 陈育才. 基于HTML5技术的移动Web前端设计与开发分析[J]. 计算机产品与流通 2019,(10),25

致 谢

本次毕业设计能够顺利完成离不开我的指导老师的指导和帮助，非常感谢我的指导教师，感谢老师在毕设各方面的指导。老师严肃的科学态度，严谨的治学精神，精益求精的工作作风，深深地感染和激励着我。老师还教导我们如何做学问和做人,其所教导的正是我所欠缺和不明白的,对我今后的人生旅途有莫大的启示。同时在老师的指导下，动手能力有所提高，相信这对马上要步入社会的毕业生是有很大的帮助的。同时在系统开发过程中许多同学都给了我很大的帮助，使我解决了不少的难点。

另外，还要特别感谢大学四年学习过程中给我系统讲授专业课知识的各位老师，感谢全班每一位同学对我的无私帮助，感谢我的指导老师彭志豪老师对我的帮助和指导，使我得以顺利完成论文。我还要感谢一直以来对我默默支持的父母。论文参阅了大量的国内外有关文献，对文献的作者也表示由衷的感谢。

时间的仓促及自身专业知识的不足，整篇论文肯定存在尚未发现的缺点和错误。恳请阅读此篇论文的老师、同学，多予指正，不胜感激。

大连东软信息学院

毕业设计（论文）原创承诺书

1、本人承诺：所提交的毕业设计（论文）是认真学习理解学校的《毕业设计（论文）工作规范》后，在教师的指导下，独立地完成了任务书中规定的内容，不弄虚作假，不抄袭别人的工作内容。

2、本人在毕业设计（论文）中引用他人的观点和研究成果，均在文中加以注释或以参考文献形式列出，对本文的研究工作做出重要贡献的个人和集体均已在文中注明。

3、在毕业设计（论文）中对侵犯任何方面知识产权的行为，由本人承担相应的法律责任。

4、本人完全了解学校关于保存、使用毕业设计（论文）的规定，即：按照学校要求提交论文和相关材料的印刷本和电子版本；同意学校保留毕业设计（论文）的复印件和电子版本，允许被查阅和借阅；学校可以采用影印、缩印或其他复制手段保存毕业设计（论文），可以公布其中的全部或部分内容。

5、本人完全了解《毕业（设计）论文工作规范》关于“学生毕业设计（论文）出现购买、他人代写、或者抄袭、剽窃等作假情形的，取消其学位申请资格；已经获得学位的，依法撤销其学位。取消学位申请资格或者撤销学位者，从处理决定之日起3年内，学校不再接受学生学位申请”的规定内容。

6、本人完全了解《学生手册》中关于在“毕业设计（论文）等环节中被认定抄袭他人成果者”不授予学士学位，并且“毕业学年因违纪受处分影响学位的学生不授予学士学位，并且无学士学位申请资格”的规定内容。

以上承诺的法律结果、不能正常毕业及其他不可预见的后果由学生本人承担！

学生本人签字：

年 月 日